mathematics

MDPI

Article

A Novel Color Image Encryption Algorithm Based on Hybrid Two-Dimensional Hyperchaos and Genetic Recombination

Yaoqun Xu [1,*], Jiaoyang Liu [2], Zelong You [2] and Tianqi Zhang [2]

[1] Institute of System Engineering, Harbin University of Commerce, Harbin 150028, China
[2] School of Computer and Information Engineering, Harbin University of Commerce, Harbin 150028, China; liujy@s.hrbcu.edu.cn (J.L.); youzelong@s.hrbcu.edu.cn (Z.Y.); zhangtianqi@s.hrbcu.edu.cn (T.Z.)
* Correspondence: xuyq@hrbcu.edu.cn

Abstract: The transition from text to images as the primary form of information transmission has recently increased the need for secure and effective encryption techniques due to the expanding information dimensions. The color picture encryption algorithm utilizing chaotic mapping is limited by a small chaotic range, unstable chaotic state, and lengthy encryption duration. This study integrates the Ackley function and the Styblinski–Tang function into a novel two-dimensional hyperchaotic map for optimization testing. A randomness test is run on the chaotic sequence created by the system to check that the new chaotic system can better sustain the chaotic state. This study introduces two techniques, genetic recombination and clock diffusion, to simultaneously disperse and mix images at the bit level. This study utilizes chaotic sequences in genetic recombination and clock drift to propose an image encryption technique. The data indicates that the method demonstrates high encryption efficiency. At the same time, the key also successfully passed the NIST randomness test, verifying its sensitivity and randomness. The algorithm's dependability has been demonstrated and can be utilized for color image encryption.

Keywords: 2D hyperchaotic map; Ackley function; Styblinski–Tang function; color image encryption; genetic recombination

MSC: 68P25

check for updates

Citation: Xu, Y.; Liu, J.; You, Z.; Zhang, T. A Novel Color Image Encryption Algorithm Based on Hybrid Two-Dimensional Hyperchaos and Genetic Recombination. *Mathematics* **2024**, *12*, 3457. https://doi.org/10.3390/math12223457

Academic Editor: Cheng-Chi Lee

Received: 4 October 2024
Revised: 28 October 2024
Accepted: 4 November 2024
Published: 5 November 2024

1. Introduction

Due to the rapid advancement of the Internet and multimedia, images are increasingly replacing words as the primary medium for transmitting information. The security of photographs has gained increasing attention, particularly with the emergence of blockchain and metaverse concepts. Since images are two-dimensional or three-dimensional data, the data correlation between neighboring pixel positions of the image is vital. Conventionally, the one-dimensional data encryption approach is no longer adequate for image encryption, so many researchers have poured into the field of encryption algorithms, and new algorithms are continually being presented. These algorithms are primarily categorized into optical-based, spatial-domain-based, and frequency-domain-based. The encryption algorithm in the spatial domain directly manipulates the image pixels using a key to ensure secure transmission. The security of the key directly impacts the outcome of encryption. Chaos is significant in this field. Edward Lorenz introduced the concept of chaotic systems in 1963 while studying meteorological systems. Claude Shannon [1] utilized it in encryption in 1994 due to its sensitivity to beginning values, randomness, determinism, and ergodicity. Claude Shannon suggested encrypting data by utilizing random sequences derived from chaotic sequences, serving as the foundation for numerous subsequent chaotic encryption techniques.

Chaos is widely used in cryptography for maintaining chaotic systems in chaotic states and generating pseudo-random sequences, making it a significant topic of research. In 1990, Aihara et al. [2] included chaotic dynamics in the Hopfield network, presenting a chaos model that addressed the issue of the network getting stuck in a locally optimal solution. Nevertheless, low-dimensional chaos mapping in image encryption has drawbacks such as a limited chaos interval, periodic window, and inadequate nonlinear dynamic behavior. Hyperchaotic systems are gaining interest in cryptography studies. Hyperchaos is characterized by many positive Lyapunov exponents in a chaotic system. This year, several strategies have been presented to address the limitations of one-dimensional chaotic mapping. Zhou et al. [3] employed a cascade approach to create a novel chaotic map by linking two one-dimensional chaotic maps sequentially. They utilized logistic mapping as a switch control to determine the iteration process between tent mapping and sine mapping. Analyzed for information entropy, a chaotic sequence exhibits a strong association. Aside from cascade and switching, there are many techniques that utilize chaotic models, including combination and coupling. Cosine and sine functions are commonly utilized to analyze the fundamental oscillation of chaotic systems simultaneously.

Recently, some new hyperchaos [4–6] have been developed using different chaos combination techniques, many of which exhibit superior performance when compared to conventional chaotic mapping. Nevertheless, there could be constraints in the execution of chaos. Its lack of sensitivity to the starting value and the volatility of the chaotic state could affect subsequent keys, making it suitable for picture encryption. There are still certain restrictions on the application. Recently, chaotic mapping design has incorporated optimized test functions [7]. This function has a more intricate visual expression compared to the sine function previously utilized. While sustaining oscillation, its oscillation ripples also exhibit increased diversity. Hence, the chaotic map derived from this function is better suited for encryption applications.

Image encryption algorithms based on chaotic systems can be divided into pixel level, bit level, and block level according to the encryption method. The pixel-level chaotic picture encryption algorithm is a widely accepted encryption technique. In 2004, Chen et al. [8] developed a three-dimensional cat map by extending a two-dimensional cat map. They rearranged the pixel coordinates of the image and applied an XOR operation using the logistic map to encrypt the image. The encrypted output is effective in repelling statistical estimates and differential attacks, but its key has a minimal correlation with the plaintext and is relatively easy to decrypt. Most encryption algorithms based on pixel level rarely deal with scrambling and diffusion strategies separately without considering the correlation between the two. Wang et al. [9] introduced a rapid picture encryption technique that utilizes diffusion and scrambling. The method involves dividing the image into pixel blocks and employing spatiotemporal chaos to create a random number sequence. This sequence is then used to simultaneously diffuse and scramble the image. Wang et al. [10] developed a novel image encryption algorithm that achieves the effects of diffusion and scrambling by decomposing the plaintext into bit planes based on its weight position and reorganizing it. The algorithm is based on Lorenz hyperchaos and genetic recombination. Recently, various encryption algorithms have been enhanced by using DNA coding [11], quantum coding [12], and other technologies, significantly diversifying encryption techniques. Despite the satisfactory performance of these algorithms in quantitative assessments, their reliance on a singular data format imposes constraints on the extension of their encryption methodologies.

A novel two-dimensional hyperchaotic map is created by utilizing the Ackley function and Styblinski–Tang function. This map is then integrated with the genetic recombination process to develop a bit-level color image encryption algorithm. Both the Ackley function and the Styblinski–Tang function are widely used optimization detection functions with multi-modal and non-concave function images. Hence, the complexity of the chaotic mapping derived from these two factors is also ensured. The picture encryption algorithm utilizes AST hyperchaos and incorporates genetic recombination to enhance confusion ability at the bit level. The 2D-AST chaos mapping and picture encryption methods have been thoroughly compared with sophisticated technologies through experiments. This study's primary contributions can be summarized as follows:

- This study introduces a novel two-dimensional chaos map with an expanded parameter range, higher Lyapunov exponent, more intricate chaotic dynamics, and improved performance in terms of permutation entropy and sample entropy.
- We utilize genetic recombination in the process of encrypting images. The new scrambling technique operates at the bit level, which has increased its resilience to attacks compared to the prior pixel-based scrambling method. We suggested a novel encryption approach that integrates several data structures and introduced a data encryption technique for genetic recombination based on the new composite framework.
- A novel color image encryption method is introduced, utilizing AST hyperchaotic mapping as the primary component and integrating the clock rotation algorithm and genetic recombination algorithm. The viability of this image encryption algorithm is validated by comparison with other existing algorithms, offering new opportunities for the development of future encryption methods.

The following sections are included in the rest of this article: The second section presents different design kinds of hyperchaotic mapping developed in recent years. The third section designs a new type of 2D hyperchaotic mapping and performs dynamic analysis on it. Section 4 introduces the clock diffusion method and genetic recombination algorithm, which are integrated with the hyperchaos from Section 3 to develop a novel color image encryption algorithm. Section 5 provides a summary of the article and discusses future prospects.

2. Existing Hyperchaotic Mapping

This section will present an overview of the chosen 2D hyperchaotic mapping [13–18]. Lin et al. [13] employed the cross-mapping approach to create a new 2D hyperchaotic map using the sine function as the infinite collapse map. This new map exhibited two positive Lyapunov exponents within its control parameter range, along with specific initial parameter values. Exhibits superior performance. Sun J [14] created a new two-dimensional hyperchaotic map by incorporating the infinite collapse map into the sine function and implementing a symmetrical structure, resulting in hyperchaos over various parameter ranges; in 2021, Hua et al. [15] created a new 2D hyperchaotic mapping by combining logistic mapping with sinusoidal mapping and increasing the phase space dimension to two dimensions. The sample entropy fluctuates about 2 despite having a broad chaotic range for its characteristics. Li et al. incorporated the sine function into the discrete mapping, resulting in complicated dynamic phenomena characterized by the oscillation of the trigonometric function. This led to a relatively stable hyperchaotic state within the specified parameter range. Qin et al. [16] initially integrated the Chebyshev map with the sine map to create a novel one-dimensional map and then incorporated the cosine map to expand its phase space to two dimensions. They utilized a coupling method to create and analyze a novel hyperchaotic system. The mapping's Lyapunov exponent remains stable in two positive states and has successfully passed the randomness test, demonstrating its suitability for picture encryption. Wang et al., 2023 [17], constructed a new 2D hyperchaotic mapping by coupling x and y using a sine function while embedding them into a logistic map, which exhibited multiple LE indices greater than 0 in multiple parameter fields. Wang et al. [18] utilized cubic mapping and infinite collapse mapping to create and analyze a novel 2D hyperchaotic mapping. The double largest Lyapunov exponent remains steady around 0, whereas the sample entropy varies at 1.6 and the permutation entropy is approximately 1. It exhibits positive, chaotic traits. In our subsequent study, we will assess the newly proposed 2D hyperchaotic map by comparing it with chaotic maps generated using other approaches. In conclusion, despite the unique strengths and weaknesses of hyperchaotic systems produced by distinct methods, they collectively enhance the intricacy and variety of chaotic dynamics, advancing our use and comprehension of chaotic systems (Table 1).

Table 1. Existing hyperchaotic mapping.

Ref.	2D Chaotic System	Parament
[13]	$\begin{cases} x_{i+1} = \sin\left(\frac{\alpha}{\sin(y_i)}\right) \\ y_{i+1} = \beta\sin(\pi(x_i + y_i)) \end{cases}$	α, β
[14]	$\begin{cases} x_{i+1} = r\sin\left(\pi\left((y_i + h)k\sin\left(\frac{a\pi}{x_i}\right)\right)\right) \\ y_{i+1} = r\sin\left(\pi\left((y_i + h)k\sin\left(\frac{a\pi}{x_i}\right)\right)\right) \end{cases}$	r, a, b, h
[15]	$\begin{cases} x_{i+1} = \cos(4ax_i(1 - x_i)) + b\sin(\pi y_i) + 1 \\ y_{i+1} = \cos(4ay_i(1 - y_i)) + b\sin(\pi x_i) + 1 \end{cases}$	a, b
[16]	$\begin{cases} x_{i+1} = \cos(\alpha\cos^{-1}(\sin(x_i - y_i))) \\ y_{i+1} = \beta\sin(\pi(x_i + y_i)) \end{cases}$	α, β
[17]	$\begin{cases} x_{i+1} = \sin(\pi((x_i + 3)\mu\sin(\pi x_i)(1 - \sin(\pi x_i)) + \theta\sin(\pi y_i))) \\ y_{i+1} = \sin(\pi((y_i + 3)\mu\sin(\pi y_i)(1 - \sin(\pi y_i)) + \theta\sin(\pi x_i))) \end{cases}$	μ, θ
[18]	$\begin{cases} x_{i+1} = \cos\left(\pi\left(ax_i^3 + (1 - a)x_i + \sin\left(\frac{a}{y_i}\right)\right) + e^{a(x_i + y_i)}\right) \\ y_{i+1} = \cos\left(\pi\left(ay_i^3 + (1 - a)y_i + \sin\left(\frac{a}{x_i}\right)\right) + e^{a(x_i + y_i)}\right) \end{cases}$	a

3. Proposed 2D-AST Mapping

3.1. Ackley Function

The Ackley function [19] is a prevalent nonlinear mathematical function introduced by David Ackley in 1987. It is now a common practice to utilize this approach for testing optimization algorithms due to its non-convexity and the presence of several local minima. The form of its function is as follows:

$$f(x) = -20 \times \exp\left(-0.2 \times \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2 \times \pi \times x_i)\right) + 20 + e, \quad (1)$$

Figure 1 displays the graph of the Ackley function in a two-dimensional space, with its lowest point located at x = y = 0. The Ackley function, comprised of cosine and exponential elements, contains numerous local minima surrounding the global minimum. It spreads outward in ripples from the global minimum point, creating a distinct pattern. The ongoing multi-modal landscape demonstrates that the Ackley function exhibits good diversity. The Ackley function can display chaotic behavior through iteration, and the resulting time series can serve as a crucial component in picture encryption techniques.

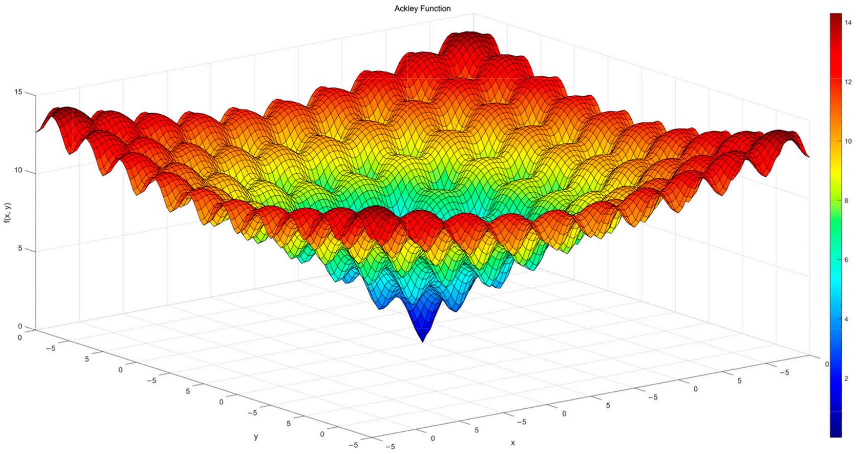


Figure 1. Ackley function.

3.2. Styblinski−Tang Function

The Styblinski−Tang function [20] was introduced by J.S. Styblinski in 1975. It is primarily utilized in performance testing optimization methods. The formula is as follows:

$$f(x) = \frac{1}{2}\sum\nolimits_{i=1}^{d}\left(x_i^4 - 16x_i^2 + 5x_i\right), \tag{2}$$

Among them, x is an n-dimensional vector, and d is the dimension of vector x. The Styblinski−Tang function is simpler than typical multi-modal functions as it is composed solely of power terms. Aside from multimodality and non-convexity, the Styblinski−Tang function also exhibits superior smoothness. Figure 2 displays a two-dimensional Styblinski−Tang function defined as follows:

$$f(x,y) = \frac{1}{2}\left[\left(x^4 - 16x^2 + 5x\right) + \left(y^4 - 16y^2 + 5y\right)\right], \tag{3}$$

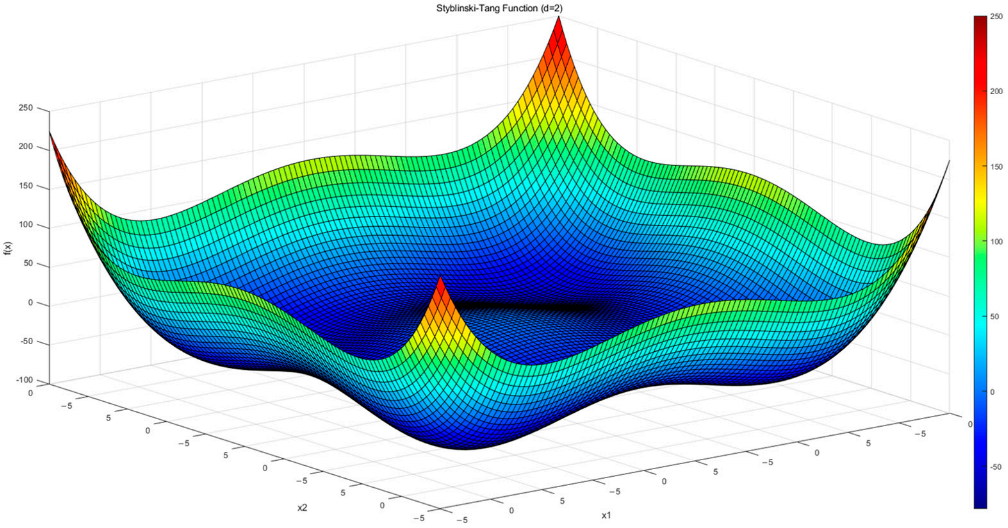


Figure 2. Styblinski−Tang function.

The global lowest point of the Styblinski−Tang function is −2.903534, which is achieved when each dimension's value is −2.903534. The number of local minimum values of a function is directly connected to its dimension, with each dimension having its local minimum. Thus, in low dimensions, the function exhibits a local minimum in each dimension while also preserving smoothness, significantly complicating identifying the optimal solution.

3.3. 2D Hyperchaotic Mapping Design

The construction of a new hyperchaotic map begins by utilizing the Ackley function as the foundation of the chaotic map. To smooth the oscillations of the chaotic map, the Styblinski−Tang function is integrated with it simultaneously. Ultimately, the mapping was restructured among the function terms to further augment its intricacy. The resultant 2D−AST chaotic mapping acquires the following form:

$$\begin{cases} x_{i+1} = -a \times \exp\left(-10 \times \sqrt{\frac{1}{2} \times \left(x_i^2 + y_i^2\right)}\right) + x_i^5 - y_i^3 + x_i \ \ mod \ \ \beta \\ y_{i+1} = -a \times \exp\left(\frac{1}{2} \times (\cos(b \times x_i) + \cos(b \times y_i))\right) + y_i^5 - x_i^3 + y_i \ \ mod \ \beta \end{cases}, \tag{4}$$

The current states of the chaotic map are denoted by x_i and y_i, the chaotic states of the subsequent time point are represented by x_{i+1} and y_{i+1}, the controllable variables of the chaotic map are a and b, and the modular operation is denoted by mod.

In Figure 3, the design path of the 2D−AST hyperchaotic mapping is illustrated. To derive the first part of the 2D−AST mapping, the two−dimensional forms of the Ackley function and the Styblinski−Tang function are divided. By combining the initial parts of both functions, the first part of the 2D−AST mapping is obtained. The second part of the mapping is derived similarly by merging the remaining parts of the Ackley function and Styblinski−Tang function.

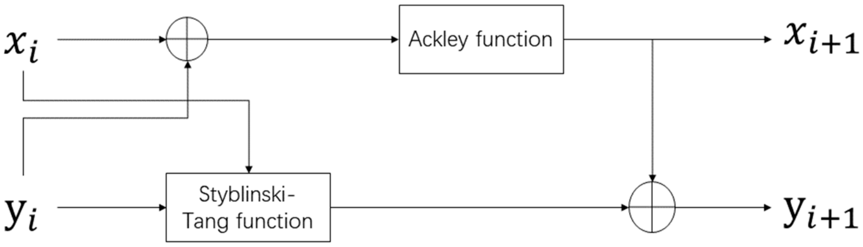


Figure 3. The design path of 2D-AST hyperchaotic mapping.

In the Ackley function, the x_i component is modified by selecting the exponential term with $x^2 + y^2$ to introduce coupling and accelerate evolution with e^a. Simultaneously, to enhance complexity, the first part of the Styblinski–Tang function, $x^4 - 16x^2 + 5x$ is incorporated. To further improve coupling and highlight its nonlinear effect, the equation is adjusted to $x^5 - y^3 + x$. The y_i component is integrated into the secondary part of Ackley's function. The exponential term of y_i involves the coupling of x and y cos() function terms, unlike x_i. The objective is to ensure the stable production of chaotic oscillatory behavior in the mapping. The y_i component retains the residual portion of the two-dimensional Styblinski–Tang function to enhance the correlation between y and x, ensuring their close interconnection and maintaining the oscillation of the cosine function throughout the iteration phase, preventing it from deteriorating or stabilizing. The specific construction process is shown in Figure 4.

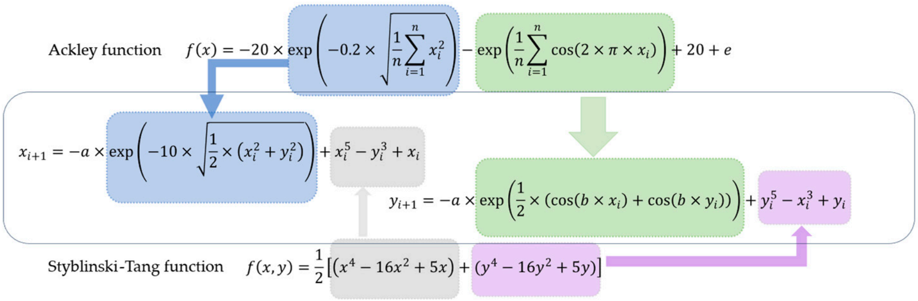
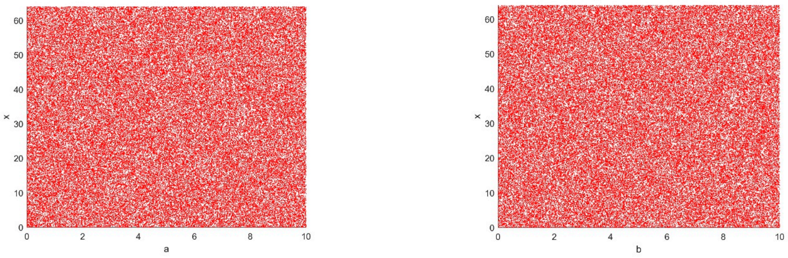

Ackley function
$f(x) = -20 \times \exp\left(-0.2 \times \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2 \times \pi \times x_i)\right) + 20 + e$
$x_{i+1} = -a \times \exp\left(-10 \times \sqrt{\frac{1}{2} \times (x_i^2 + y_i^2)}\right) + x_i^5 - y_i^3 + x_i$
$y_{i+1} = -a \times \exp\left(\frac{1}{2} \times (\cos(b \times x_i) + \cos(b \times y_i))\right) + y_i^5 - x_i^3 + y_i$
Styblinski-Tang function
$f(x,y) = \frac{1}{2}[(x^4 - 16x^2 + 5x) + (y^4 - 16y^2 + 5y)]$

Figure 4. Mathematical construction of 2D-AST hyperchaotic mapping.

Prevent it from deteriorating or stabilizing. The 2D-AST chaos map incorporates power terms and introduces the mod function to enhance the observation of chaotic behavior. The 2D hyperchaotic map is defined by three parameters: a, b, and β, where a and b are within the range of (0, ∞) to sustain a chaotic state. The suitability of a chaotic map as an encryption key for images is contingent upon the parameters' range. The key's complexity is defined by the range of its parameters. The AST chaotic map remains sensitive to the starting parameters and is extremely flexible to them. This feature enables the generation of diverse chaotic sequences based on random factors, enhancing the complexity of the key and improving resistance against brute force attacks. Thus, the chaotic mapping is suitable for use in encryption.

3.4. Bifurcation Diagrams and Trajectories

Bifurcation diagrams are a crucial tool for analyzing chaotic mapping and serve as a widely used visual assessment method. The bifurcation diagram illustrates the evolution of each component in the chaotic map when the parameters are manipulated, depicting the map's shift from order to chaos. A high-quality chaotic map should display a scattered and evenly distributed bifurcation diagram. Reducing the number of trajectories shown in the bifurcation diagram improves the visualization of chaos in the mapping. Figure 5 displays the bifurcation diagram. By adjusting the parameters a and b, it is observed that the scatter plots of the chaotic map exhibit consistent spacing and lack a clear trajectory, suggesting that the 2D hyperchaotic map possesses distinct features under each parameter setting. This implies that the chaotic map has a broader interval, meeting the encryption algorithm's essential criteria for enhanced chaos properties.

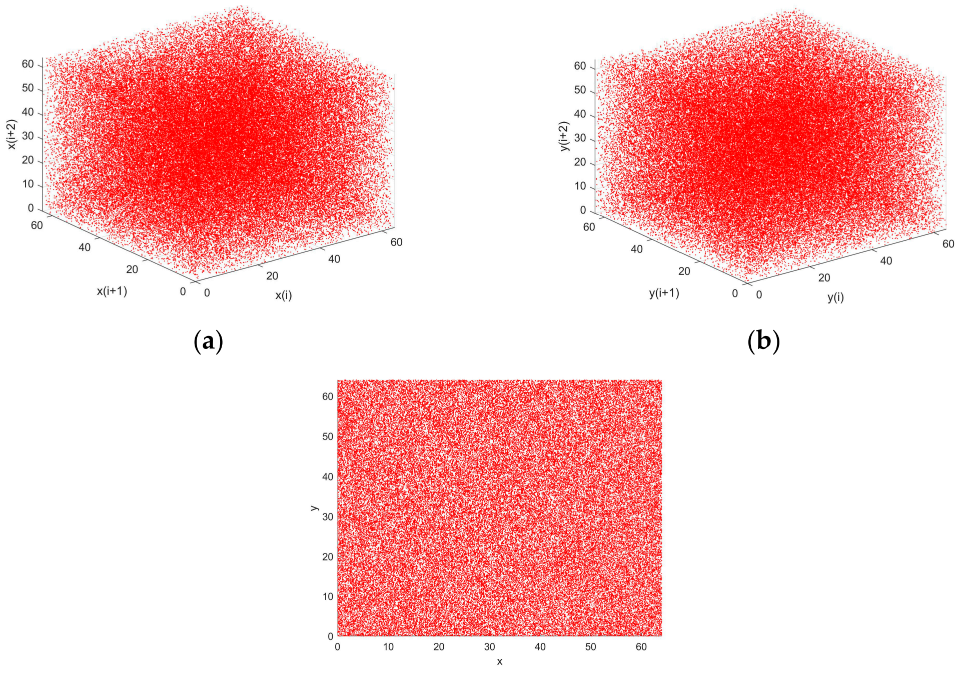


(**a**) (**b**)

Figure 5. Bifurcation diagrams, (**a**) x_i *and a*; (**b**) x_i *and b*.

.

Phase space trajectory is a frequently utilized tool in chaos mapping studies. It depicts the system's state at each moment as a point in phase space. The point's trajectory represents the system's development as time progresses. In a chaotic system, the anticipated system path is dispersed and spread out in phase space, lacking any discernible pattern. Figure 6 displays the two-dimensional spatial trajectory diagrams of x_i and y_i, as well as the three-dimensional trajectory diagrams of x_i, x_{i+1} and x_{i+2}; y_i, y_{i+1} and y_{i+2}. No clear paths are visible in the three photos depicting trajectories. This is closely linked to the improved chaotic behavior of the AST chaos map. The 2D hyperchaotic map demonstrates significant sensitivity to the beginning value and can be utilized in the field of encryption.



(**a**) (**b**)

(**c**)

Figure 6. Phase space trajectory: (**a**) x_i, x_{i+1} and x_{i+2}; (**b**) y_i, y_{i+1} and y_{i+2}; (**c**) x_i, y_i.

3.5. Lyapunov Exponent

The Lyapunov exponent (LE) [21] is the primary metric used to assess the presence of chaos in a system. The essence is to quantify the departure rate between neighboring trajectories of the system in phase space. If the LE is positive, small disturbances in the system will rapidly grow, causing the trajectory to spread out. The system is currently chaotic, characterized by great sensitivity to initial conditions and considerable unpredictability. On the contrary, it signifies that the current system is in a stable state and will not be disturbed by tiny changes in the original value. Hyperchaos refers to the condition in which multiple positive LEs of a high-dimensional system exist in the same state. In contrast to low-dimensional chaotic mapping, hyperchaotic systems exhibit increased complexity, accelerated evolution, and heightened sensitivity.

The computation of the multiple LEs of a multidimensional chaotic system is accomplished by utilizing the Jacobian matrix $J(x_i, y_i)$ of the system. The matrix formula for a chaotic mapping in two dimensions is given below:

$$J(x_i, y_i) = \left(\left(\begin{matrix} \left(\frac{\partial f_1(x_i,y_i)}{\partial x_i}\right) \\ \left(\frac{\partial f_2(x_i,y_i)}{\partial x_i}\right) \end{matrix}\right)\left(\begin{matrix} \left(\frac{\partial f_1(x_i,y_i)}{\partial y_i}\right) \\ \left(\frac{\partial f_2(x_i,y_i)}{\partial y_i}\right) \end{matrix}\right)\right), \tag{5}$$

Currently, the chaos mapping equation is as follows:

$$f(x, y) = \begin{cases} x_{i+1} = f_1(x_i, y_i) \\ y_{i+1} = f_2(x_i, y_i) \end{cases}, \tag{6}$$

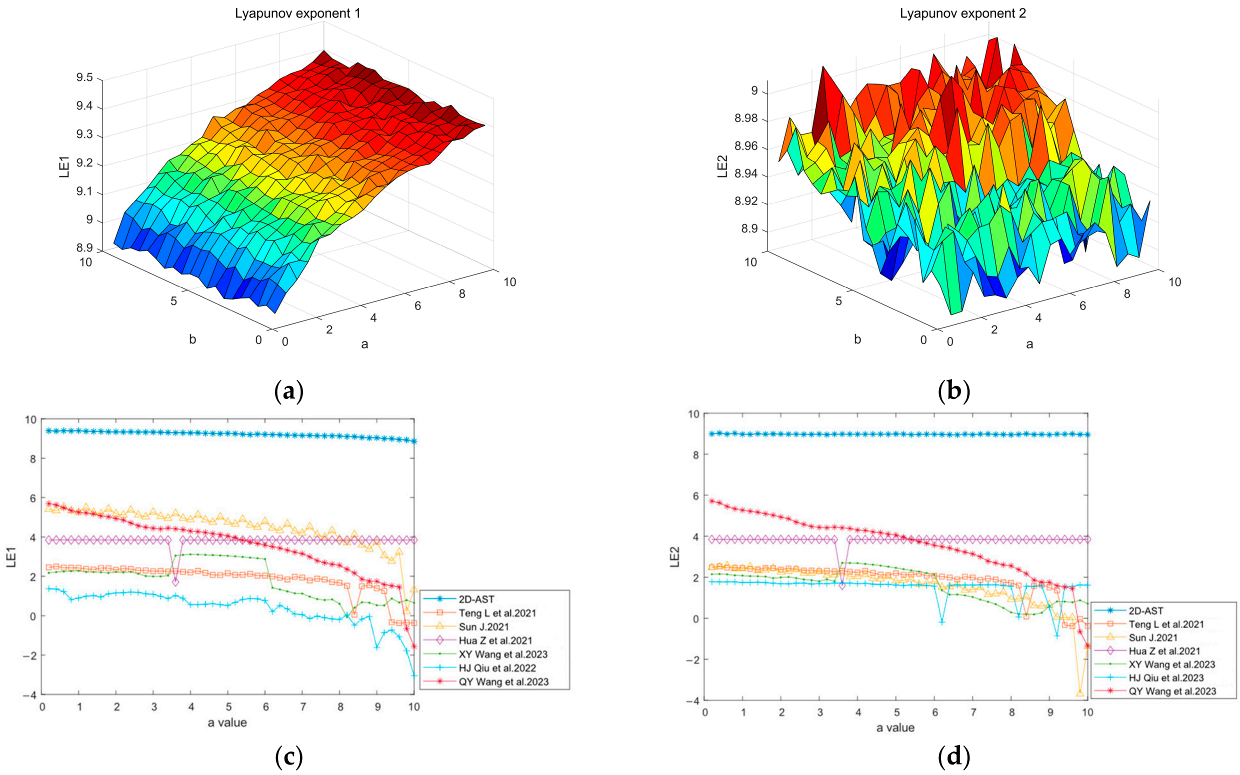
The Lyapunov calculation formula is:

$$LE_i = \lim_{n \to \infty} \frac{1}{n} \sum\nolimits_{i=1}^{n-1} \ln|\lambda_i(J)|, \tag{7}$$

$\lambda_i(J)$ represents the i-th eigenvalue of the Jacobian matrix $J(x_i, y_i)$, while n denotes the number of iterations that can be performed. Lyapunov exponent 1(LE1) and Lyapunov exponent 2(LE2) are the LEs of x and y in the 2D chaotic map, respectively. LE1 and LE2 of the two-dimensional AST chaos map have a range of chaos values between 0 and ∞. The Figure 7 depicts the three-dimensional representation that occurs when the a and b control parameters are altered. It is apparent from the image that when the control parameters are altered within the range of 0 to 10, LE1 and LE2 also experience minor fluctuations; however, they remain essentially constant and exhibit significantly higher values. is equal to zero, proving that the 2D-AST chaotic map is hyperchaotic. Simultaneously, the parameters β = 8, b = 10, and a ∈ (0, 10) are chosen in order to validate the AST map compared to the pre-existing two-dimensional chaotic map. Table 2. presents the values of the items. The respective average Lyapunov indices for 2D-AST are 9.2184 and 8.9975. 4.4839 and 3.7984 are the nearest values of alternative chaotic maps, both of which are inferior to the LE of AST. Consequently, the chaotic map presented in this paper maintains a hyperchaotic state at a ∈ (0, 10), demonstrating that it exhibits commendable chaotic performance.

Table 2. Comparison of chaos mapping indicators.

Chaos	LE1	LE2	PE	SE
[13]	1.9	1.8481	0.96811	1.3982
[14]	4.4839	1.549	0.99383	1.807
[15]	3.8005	3.7984	0.99865	0.90858
[16]	1.844	1.6131	0.97261	1.3408
[17]	0.37197	1.5189	0.85285	0.49676
[18]	3.5989	3.6037	0.98177	1.592
AST	9.2184	8.9775	0.99875	2.0947



(**a**) (**b**)

(**c**) (**d**)

Figure 7. (**a**,**b**): 2D-AST chaos map Lyapunov exponent diagram; (**c**,**d**): Lyapunov exponent comparison chart [13–18].

3.6. Sample Entropy

Proposed by Richman et al., in 2000, sample entropy (SE) [22] is a statistical index that assesses time complexity and regularity. Approximate entropy [23] is utilized to enhance the entropy of the sample. The complexity of the time series is quantified by both metrics, with the dimension change representing the likelihood that the sequence produces novel patterns. In contrast to alternative metrics utilized to assess chaos dynamics, sample entropy exhibits a more robust characteristic. A comparison chart of sample entropy under various parameters is presented in Figure 8. As the SE increases, the chaotic sequence's regularity decreases, increasing the sequence's complexity.

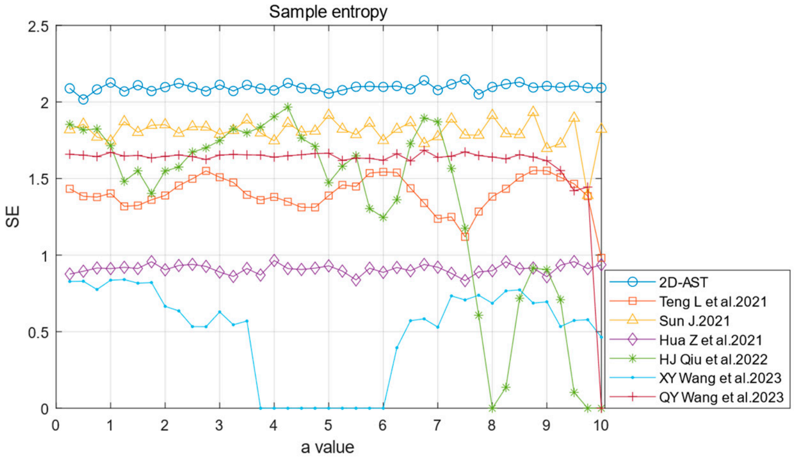


Figure 8. Sample entropy [13–18].

The image reveals that the sample entropy of the chaotic map fluctuates to varying degrees in response to parameter changes, whereas the fluctuation of the AST chaotic map is relatively modest and remains consistently above 2. The mean sample entropy of every chaotic map within the parameter range of 0 to 10 is presented in Table 2. The 2D-AST chaotic map possesses a number of benefits in comparison to the current 2D hyperchaos.

3.7. Permutation Entropy

Permutation entropy serves as a statistical metric for both time complexity and regularity. In contrast to sample entropy, permutation entropy [24] computes the permutation frequencies of distinct subsequences after establishing a fixed-length window and partitioning the time series into non-overlapping subsequences. Since identical numbers arranged differently will be identified as distinct, this method is more sensitive than sample entropy. The degree of randomization of the sequence is denoted by PE. When the entropy value [25] is greater, the time series becomes more intricate and stochastic. Conversely, this indicates that the time series norms are straightforward. The comparison of the permutation entropies of the chaotic map and 2D-AST, as presented in Table 2, is illustrated in Figure 9. As the permutation entropy approaches 1, it signifies an increase in complexity and an enhancement in chaotic performance. It can be seen from the image that the permutation entropy of the AST chaotic map does not change substantially as the parameters change in (0, 10) and can be stabilized at 1. Several chaotic maps have exhibited a substantial reduction in the range of specific parameters; however, there are still some chaotic maps in which the distinction from the image is difficult to discern. The mean value of the permutation entropy for every chaotic map is presented in Table 2. It is evident that while numerous chaotic maps exhibit permutation entropies exceeding 0.99, the AST permutation entropy remains marginally greater than that of other chaotic maps at 0.0015. This suggests that while the parameters of AST may fluctuate, their permutation entropy remains at an optimal level.

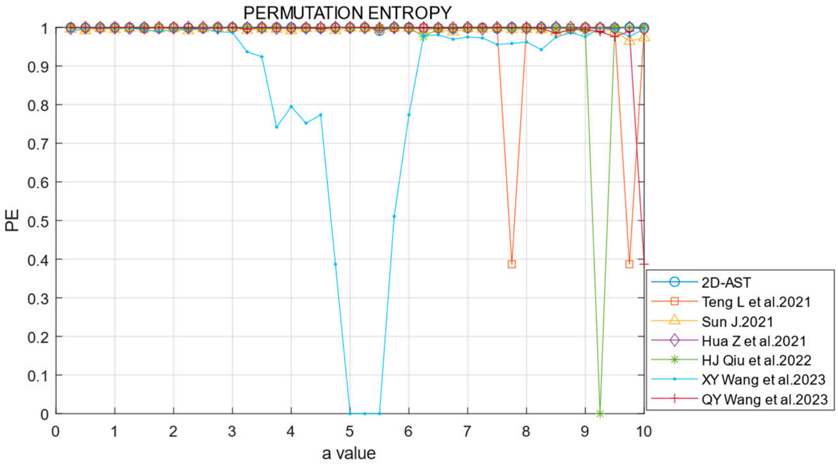


Figure 9. Permutation entropy [13–18].

3.8. Cobweb Plot

In the examination of nonlinear dynamic systems, the Cobweb plot serves as a visualization technique frequently employed to study its iterative course. A qualifying chaotic system inside a dynamic framework must have a trajectory in a chaotic state that is discrete, non-repetitive, and does not converge to one or multiple discrete points as iterations increase. Figure 10 illustrates the Cobweb plot of x and y for the AST chaotic map, the intersection point in the graph is its possible convergence point. The graphic clearly illustrates that as the number of iterations increases, the trajectories become uniformly distributed, and the output values do not exhibit a convergence trend, demonstrating the system's robust chaotic properties.

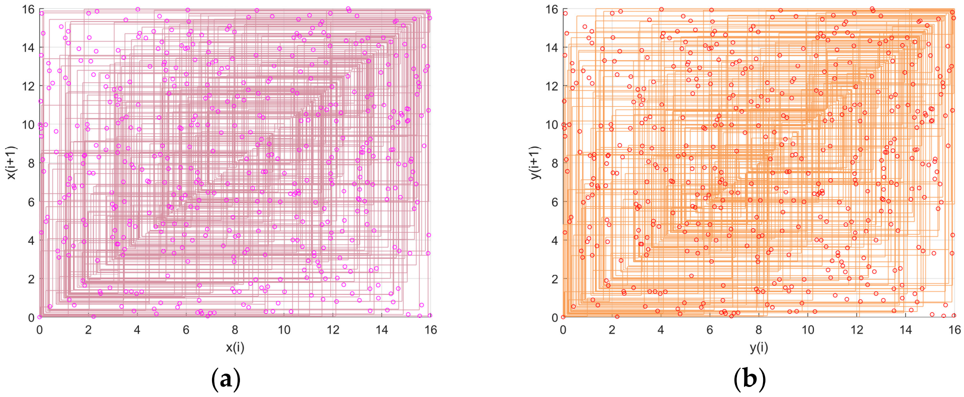


Figure 10. Cobweb Plot, (**a**) x_i *and* x_{i+1}; (**b**) y_i *and* y_{i+1}.

3.9. NIST Randomness Test

If the chaotic sequence is utilized in the field of encryption, its randomness is a crucial reference index. The NIST SP800-22 statistical test standard [21], developed by the American National Institute of Standards and Technology (NIST) Institute and comprising 15 statistical packages, is one of the most prevalent random test experiments. In this paper, the initial parameters x = 0.5, y = 0.5, a = 10, b = 10, and 100,000,000 iterations are conducted to process the obtained chaotic sequence. The formula for processing is shown in Equations (8) and (9), and the processed time series will be tested for randomness, and the results of the test are displayed in Table 3.

$$y_i = \left(10^{10} \times x_i\right)\ mod\ 1, \tag{8}$$

$$z(t) = \begin{cases} 0,\ 0 \le y_i < 0.5 \\ 1,\ 0.5 \le y_i < 1' \end{cases} \tag{9}$$

Table 3. NIST result.

Subset	x		y	
	***p*-Value**	**Proportion**	***p*-Value**	**Proportion**
Frequency	0.596578	99%	0.109898	100%
Block frequency	0.879688	98%	0.537979	100%
Cumulative sums	0.754420	100%	0.691823	100%
Runs	0.474986	99%	0.817415	98%
Longest run	0.621999	99%	0.054168	99%
Rank	0.455937	100%	0.869096	99%
FFT	0.883171	98%	0.153024	100%
Non-overlapping	0.595549	100%	0.799433	98%
Overlapping	0.867692	99%	0.114662	100%
Universal	0.419021	98%	0.471428	99%
Approximate entropy	0.394250	100%	0.274153	100%
Random excursions	0.941144	100%	0.899947	98%
Random excursions variant	0.957319	100%	0.612846	100%
Serial	0.419021	99%	0.592468	100%
Linear complexity	0.401199	100%	0.463541	100%

The NIST randomness passing criterion is that the *p*-value is greater than 0.01, and the 15 test sets in the table are all greater than 0.01, so it can be demonstrated that the random sequence is random and can be applied to the field of encryption.

4. Proposed Encryption Algorithm

The picture encryption algorithm utilizing 2D-AST hyperchaotic mapping is often segmented into three components: key generation, clock diffusion, and genetic recombination. The initial sub-algorithm is a key generation, which acquires the starting parameters of the chaotic map from the original image to enable the AST map to create the encryption key. The second part encrypts the pixel values of the image's three channels using clock diffusion and the generated key. The final sub-algorithm employs a genetic recombination technique to randomize the pixels at the bit level.

4.1. Key Generation

Keys are always crucial in image encryption methods. The complexity of an encryption algorithm's key directly impacts the security of the image encryption algorithm. This article's encryption approach utilizes the time series of the AST chaotic map as the key. The chaotic map is highly sensitive to the beginning value in the chaotic state; hence, the initial value of the generated key comes from the initial image. Both the clock technique and the genetic recombination procedure necessitate keys during the subsequent encryption process. To guarantee the security of the encryption algorithm, chaotic keys with varying initial values will be employed at different encryption stages. This study uses SHA-512 to compute the hash value $H = (h_1, h_2, \cdots, h_{64})$ derived from the plaintext picture to establish the initial value. The precise procedure is outlined as follows:

$$H' = hex2dec(H), \tag{10}$$

$$\begin{cases} x_1 = mod(H'(5) \times H'(9), 15) + (H'(1) + H'(13)) \times 2^{-8} \\ y_1 = mod(H'(18) \times H'(22), 15) + (H'(14) + H'(24)) \times 2^{-8} \\ a_1 = mod(\min(H'(25), \cdots H'(33)) \times \max(H'(34), \cdots H'(42)), 10) \\ b_1 = mod(\min(H'(43), \cdots H'(51)) \times \max(H'(52), \cdots H'(60)), 10) \end{cases} \tag{11}$$

$$\begin{cases} x_2 = mod(H'(6) \times H'(10), 15) + (H'(2) + H'(14)) \times 2^{-8} \\ y_2 = mod(H'(19) \times H'(23), 15) + (H'(15) + H'(25)) \times 2^{-8} \\ a_2 = mod(\min(H'(26), \cdots H'(34)) \times \max(H'(35), \cdots H'(43)), 10) \\ b_2 = mod(\min(H'(44), \cdots H'(52)) \times \max(H'(53), \cdots H'(61)), 10) \end{cases} \tag{12}$$

$$\begin{cases} x_3 = mod(H'(7) \times H'(11), 15) + (H'(3) + H'(15)) \times 2^{-8} \\ y_3 = mod(H'(20) \times H'(24), 15) + (H'(16) + H'(26)) \times 2^{-8} \\ a_3 = mod(\min(H'(27), \cdots H'(35)) \times \max(H'(36), \cdots H'(44)), 10) \\ b_3 = mod(\min(H'(45), \cdots H'(53)) \times \max(H'(54), \cdots H'(62)), 10) \end{cases} \tag{13}$$

The hex2dec() function transforms the hash value from hexadecimal to decimal. Three chaotic time series are needed for the encryption procedure of this article. Consequently, for the identical plaintext, three distinct sets of x, y, a, and b must be constructed for temporal diffusion and genetic recombination, respectively. Algorithm 1 utilizes initial values x and y, along with control parameters a and b, to employ the AST chaos map as the iteration core for producing chaotic times x_i and y_i.

The chaotic time series x and y vary in range according to the value of β. In the following research of this article, the value of β is chosen to be 8. The current chaotic sequence is a floating-point number less than β. The chaotic time series will be preprocessed to facilitate its use in the encryption process. The preprocessing formula is as follows:

$$q = \left\lfloor (x_i - floor(x_i)) \times 10^{14} \right\rfloor mod\ n, \tag{14}$$

x_i represents the chaotic time series, n stands for the number of operations needed in the encryption procedure, and q is the encryption key for the final image.

Algorithm 1. Key generation

```
Input: $x_1$, $y_1$, a, b, k, $\beta$;
Output: $x$, $y$
Start
Get $x_1$, $y_1$, a, b
for i = 1:k
    $x_{i+1} \leftarrow -a \times \exp\left(-10 \times \sqrt{\frac{1}{2} \times \left(x_i^2 + y_i^2\right)}\right) + x_i^5 - y_i^3 + x_i \ \ mod \ \ \beta;$
    $y_{i+1} \leftarrow -\exp\left(\frac{1}{2} \times (\cos(b \times x_i) + \cos(b \times y_i))\right) + y_i^5 - x_i^3 + y_i \ \ mod \ \beta;$
End
Stop
```

4.2. Clock Spreading Algorithm

Diffusion and scrambling are key encryption techniques in traditional picture encryption systems. In recent years, several diffusion encryption methods, like DNA encoding, have been commonly utilized in the diffusion stage. Their performance at the diffusion stage is commendable. Due to the encoding and decoding procedure, it has some drawbacks in terms of computing time. Hence, we suggest a novel diffusion encoding technique grounded on the concept of the clock. The clock's hour hand, minute hand, and second hand are represented by R, G, and B, respectively. The rotation amplitude is determined using a key to encrypt pixel values. Its pseudocode is displayed in Algorithm 2.

Algorithm 2. Clock spreading algorithm

```
Input: $x_1$, I; $x_1$ is the chaotic time series produced by AST chaos mapping, and I is the image to be encrypted.
Output: $x$, $I_{ciped}$
1. Start
2. Get $x$, $I$,
3. [M, N] = size ($I$);
4. $I_r$ = $I$(:,:,1); $I_g$ = $I$(:,:,2); $I_b$ = $I$(:,:,3);
5. $Key_r \leftarrow (x - floor(x)) \times 10^{14};\ Key_g \leftarrow \lfloor Key_r/60 \rfloor;\ Key_b \leftarrow \left\lfloor Key_g/60 \right\rfloor;$
6. $I_r \leftarrow mod(I_r + Key_r, 256);\ I_g \leftarrow mod\left(I_g + Key_g, 256\right);\ I_g \leftarrow mod\left(I_g + Key_g, 256\right);$
7. End
8. Stop
```

The simulation process of clock diffusion is illustrated in Figure 11. Upon analyzing the image, the following three channels are acquired: R, G, and B. The code lines containing the necessary keys for the diffusion procedure are 4. To minimize gray value correlation among the three channels of a single pixel, the following is implemented: the R channel represents the second hand, the G channel represents the minute hand, and the B channel represents the hour hand. The R channel consistently undergoes a 360-unit rotation, the G channel undergoes a 60-unit rotation, and the B channel undergoes a one-unit rotation. Additionally, since the grayscale value range is [0, 255], the mod() function will be applied subsequent to the rotation process in order to restrict the range of pixel values to a specified region. In summary, this pseudocode executes the clock diffusion algorithm, which encodes pixel values after converting the incoming chaotic sequence into the encryption key of the corresponding channel after operation, and each channel of the image is entered into the corresponding vector to facilitate subsequent operations.

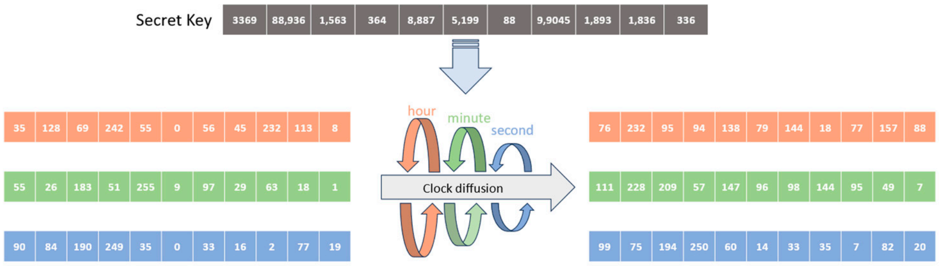


Figure 11. Clock spreading algorithm.

4.3. Genetic Recombination Algorithm

The rearrangement of DNA sequences via the separation and reassembly of chromosomes or chromosome segments is referred to as genetic recombination. Genetic recombination [26] is the exchange of genetic material between distinct organisms, which results in progeny with a unique combination of characteristics compared to their parents. Genome recombination is a naturally occurring process that expands the genetic diversity of sexually reproducing organisms, thereby enabling them to manifest novel phenotypes. Moreover, each pixel can be represented by an 8-bit binary, and the weights and quantities of information associated with data at various positions vary. The decomposed image of coffee's photograph is depicted in Figure 12.

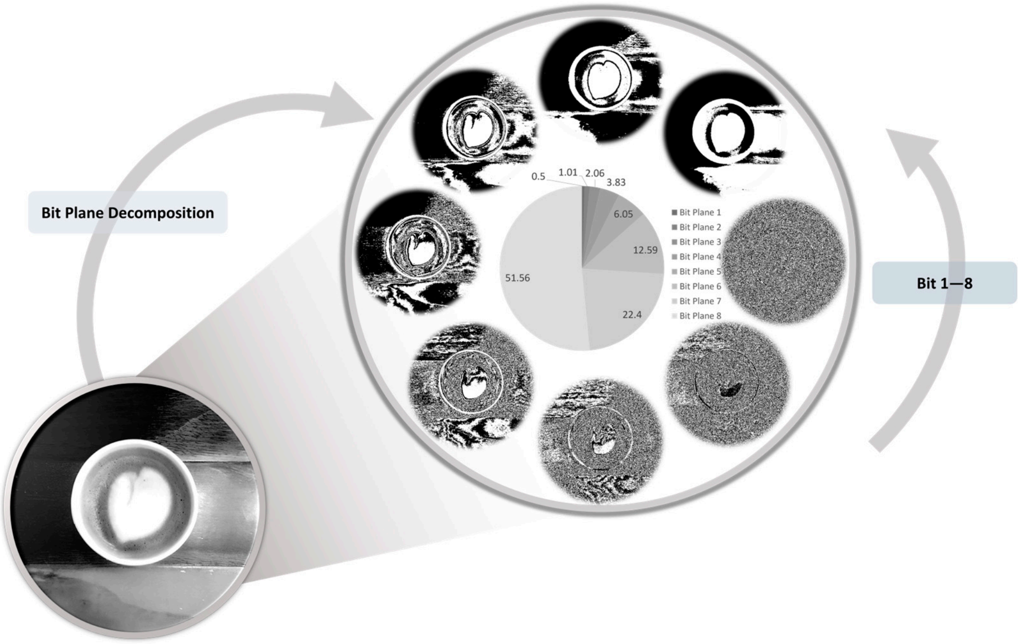


Figure 12. Exploded image of coffee picture.

As illustrated in the figure, the R, G, and B channels of the image are initially decomposed. It is evident that as the bit plane decreases, there is a corresponding gradual reduction in the quantity of information contained within the image. Consequently, to achieve a random distribution of bits conveying disparate information within the encrypted image, this work architecturally divides the pixel bits and converts them into queues and stacks. Each process sequentially selects distinct pixel blocks and recombines them based on the key. The algorithm is shown in Algorithm 3. Subsequently, the article performs the following operations: it swaps the high and low three bits of the R channel, exchanges the high and low three bits of the G channel, and finally advances the lowest bit of the B channel to the highest bit. The resulting image retains its original dimensions, but it is evident that a substantial amount of information has been lost. Acquiring corresponding information from images after genetic recombination poses a significant change. The second pseudocode is illustrated in Algorithm 3.

Algorithm 3. Genetic recombination algorithm

Input: a, b are two pixels selected for gene recombination; $index$ is the breakpoint of this gene recombination.
Output: I_{ciped}

```
1.  Start
2.  Get $a, b, index$;
3.  $binary\_a \leftarrow dec2bin(a, 8)$;
4.  $binary\_b \leftarrow dec2bin(b, 8)$;
5.  $queue\_c$ = []; $stack\_c$ = [];
6.  $for\ j = 1 : i$    $queue\_c \leftarrow [queue\_c, binary\_a(j)]$
7.  $for\ j = i + 1 : 8$    $stack\_c \leftarrow [binary\_b(j), stack\_c\ ]$
8.  $d = []$;
9.  $for\ j = 1 : length(queue\_c)$    $d \leftarrow [d, queue\_c(j)]$
10. $for\ j = 1 : length(stack\_c)$    $d \leftarrow [d, stack\_c(j)]$
11. $d = bin2dec(d)$;
12. End
13. Display $I$
14. Stop
```

Dec2bin() and bin2dec() in pseudocode 3–4 and 11 are functions that facilitate the conversion of pixel values between decimal and binary formats. The parameter c denotes the storage parameter of the algorithmic process, categorized as *stack_c* in stack format and *queue_c* in queue format. The operational procedure is illustrated in Figure 13. The first and tenth pixels in the plaintext are designated as target factors for recombination via the key, with the chosen recombination point being five. “10000101” and “00101101” represent 8-bit pixel values, respectively. The data are read sequentially from the highest bit and partitioned based on bit 5. Queue Q1 receives “001”, stack S1 receives “10110”, queue Q2 receives “001”, and stack S2 receives “10100”. Utilizing the first-in-first-out principle of queues and the first-in-last-out principle of stacks, we amalgamate the head of the queue with the base of the stack to create composite structures H1 and H2 that maintain identical output sequences. H1 and H2 are sequentially output, and the final decimal results “44” and “105” are reintegrated into the sequence to finalize gene recombination.

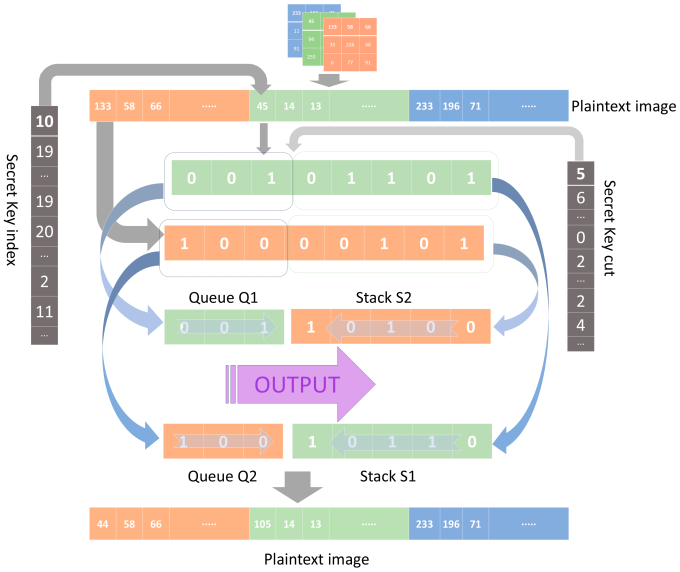


Figure 13. Genetic recombination algorithm.

4.4. Image Encryption Algorithm

Figure 14 illustrates the execution of a color image encryption algorithm that employs genetic recombination and clock diffusion.

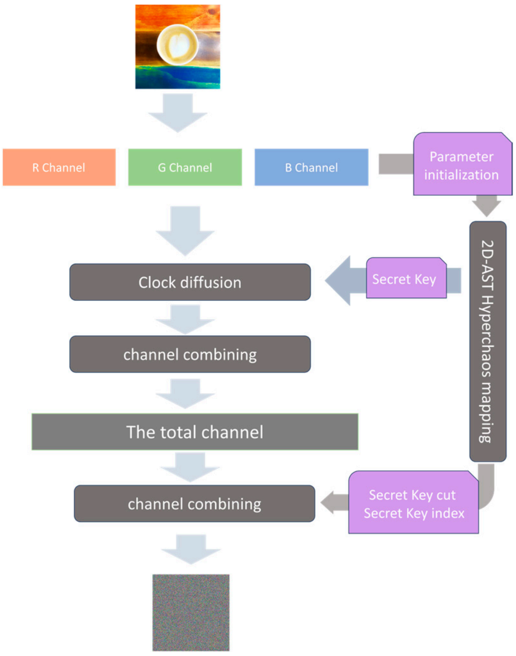


Figure 14. Image encryption process.

Step 1. Input the M × N × 3 color image into three matrices denoted as I_r, I_g and I_b, utilizing the three channels of R, G, and B, respectively. The elements and grayscale values of the corresponding channel pixels in all three matrices are identical.

Step 2. The clock diffusion encryption process necessitates a single key sequence, whereas the succeeding genetic recombination process demands two key sequences. As a result, the encryption procedure described in this study necessitates a total of three key sequences so as to maximize the degree of equality between keys. Three disorganized time series constitute independent key sequences. As initial parameters, we select the fourth and sixth planes of I_r, the seventh and fourth planes of I_g, and the fifth and second planes of I_b, respectively; that is, we obtain three initial x_i (i = 1, 2, 3) by applying Formula (5) to a = 40, 72 and 18.

Step 3. Pass x_i (i = 1, 2, 3) as the initial parameter into the 2D-AST chaos mapping so that it iterates M × N + 100 and M × N × 3 + 1000, respectively, and removes the first 1000 data to ensure the chaos of its time series, and finally obtains two time series with length M × N × 3 and a chaotic time series with length M × N.

Step 4. In step three, chaotic time series of the M × N type are processed using Formula (6). At this moment, the value $n = 1$ is chosen. The clock key of the R channel is the processed one-dimensional sequence, which is compared to 60 and 360, respectively. By performing a division operation, the keys of G and B can be obtained. The acquired key is appended to the corresponding channel and maintained within its valid range using the mod() function. The three newly acquired vectors I_r, I_g, and I_b represent the encrypted pixels.

Step 5. Formula (6) is applied to the M × N × 3 chaotic time series acquired in step 3. The values n = M × N × 3 and 8 are subsequently chosen to represent the breakpoint coordinates and index coordinates of genetic recombination, respectively. Convert the three vectors I_r, I_g, and I_b to unit8 form and combine them into a one-dimensional vector. To obtain a new one-dimensional vector I, select the exchange point using the index key, break the link, and reorganize the breakpoint key. The final encrypted sequence is as follows.

Step 6. The one-dimensional vector I should be converted to a decimal matrix of size M × N × 3 before being output. This matrix represents the encrypted final image.

The process of decrypting the image is the exact opposite of the encryption operation.

5. The Experiments for the Proposed Encryption Scheme

This section chooses three color photographs of coffee, an airplane, and a baboon as experimental objects. Airplane and baboon are from the SIPI public database. Their respective dimensions are 512 × 512 × 3, 512 × 512 × 3, and 256 × 256 × 3, unless otherwise specified. The chosen parameters for 2D-AST are $\beta = 8$ and $a = b = 10$.

5.1. The Correctness Analysis of the Algorithm

The image encryption algorithm must meet these two conditions in order to ensure the secure transmission of information, which is its primary function. One is that the image must become so cluttered that valid information cannot be extracted following encryption algorithm processing. The second requirement is that the encrypted file can be decrypted. Obtain the information in its entirety prior to encryption. The encryption and decryption outcomes of three images are illustrated in Figure 15.

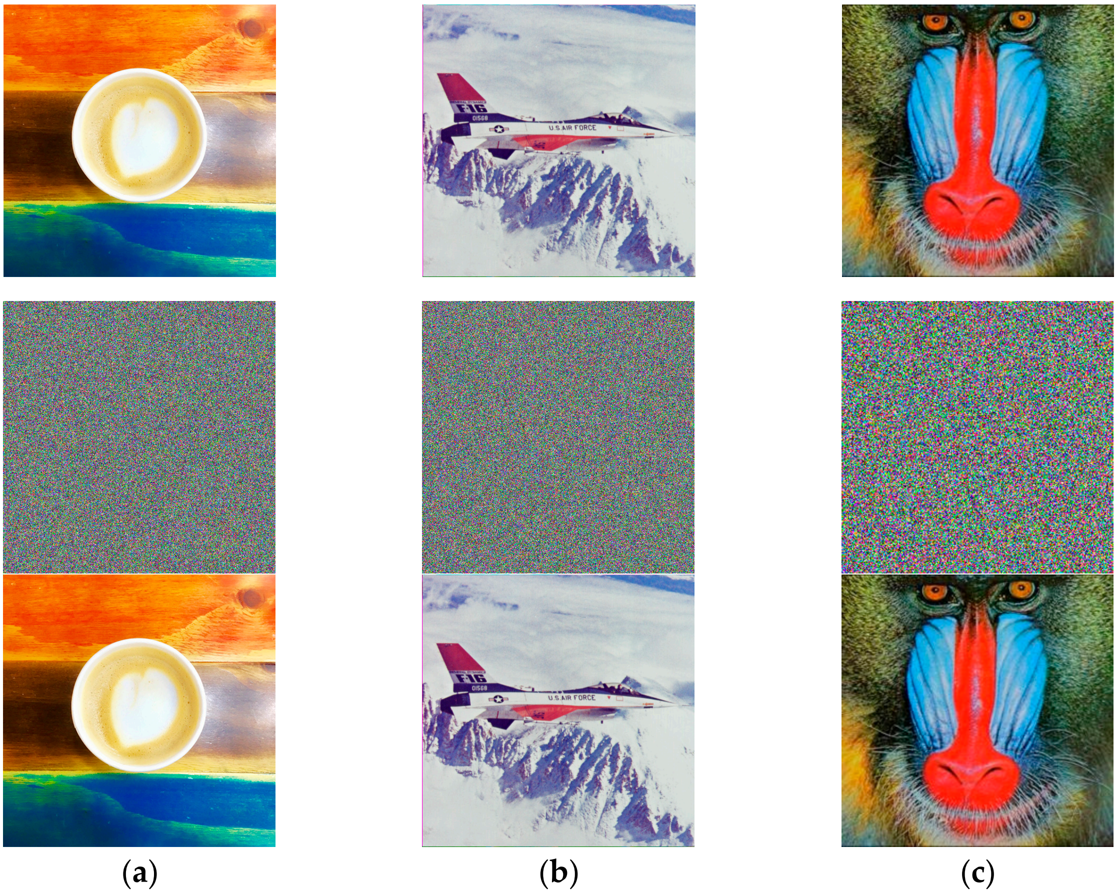


Figure 15. Encryption and decryption comparison of test objects: (**a**) coffee (512 × 512 × 3); (**b**) airplane (512 × 512 × 3); (**c**) baboon (256 × 256 × 3).

Figure 15 demonstrates that it is difficult for human vision to extract useful information from an encrypted image. This algorithm can accurately encrypt and decrypt images since the decrypted image retains all of the information contained in the original image.

5.2. Information Entropy

Information entropy [27] is the mean quantity of information contained in each received message, and a decent encrypted image should have sufficient randomness. The formula for the information entropy is:

$$Q(x) = \sum_{i=0}^{n} R(x_i) log_2 \frac{1}{R(x_i)}, \tag{15}$$

R(xi) represents the proportion of gray pixel distribution within an image, and n is the number of different pixel values, the total number of values that pixel value x can take. Theoretically, a reasonable entropy value for encrypted image data should be close to 8. Tables 4 and 5 depict the information entropy of various password images of the encryption algorithm and the information entropy comparison result. According to Table 5, the information entropy of a single channel of a 512 × 512 × 3 encrypted image is kept above 7.999, and the information entropy of its RGB three channels is 7.9998. The information entropy of a 256 × 256 × 3 image is stable above 7.9970, and the information entropy of RGB reaches 7.9990. This indicates that the pixels in these encrypted images are distributed uniformly and randomly; their information is disordered.

Table 4. Information entropy result.

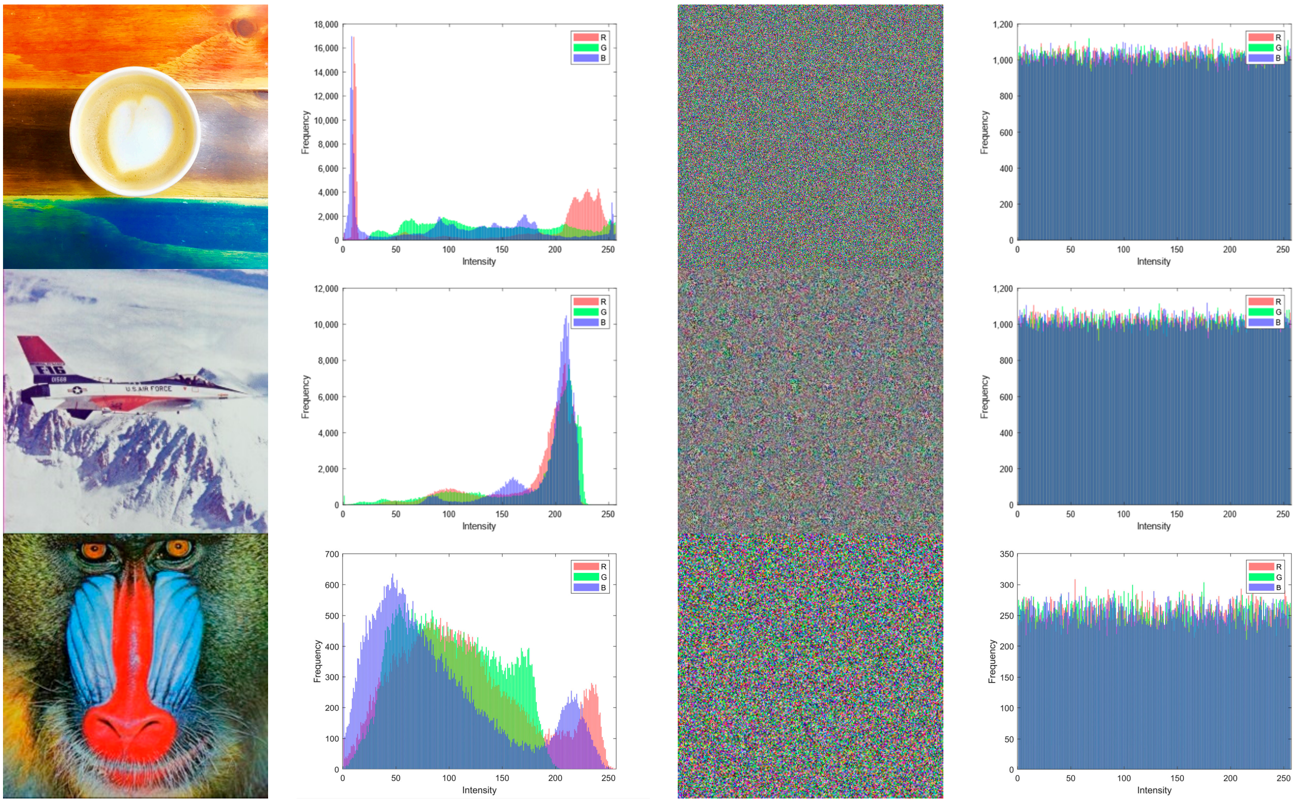
Image	Cipped R	Cipped G	Cipped B	Cipped RGB
Coffee-512	7.9994	7.9993	7.9993	7.9998
Airplane-512	7.9993	7.9993	7.9993	7.9998
Baboon-256	7.9972	7.9970	7.9970	7.9990

Table 5. Information entropy comparison.

Image	File	Cipped R	Cipped G	Cipped B
Lena-512	This paper	7.9993	7.9994	7.9994
	[28]	7.9992	7.9993	7.9993
	[29]	7.9993	7.9994	7.9993
	[30]	7.9975	7.9974	7.9973
Airplane-512	This paper	7.9993	7.9993	7.9993
Baboon-256	This paper	7.9972	7.9970	7.9970
Lena-256	This paper	7.9971	7.9970	7.9970
	[30]	7.9969	7.9971	7.9971
	[29]	7.9968	7.9973	7.9969

5.3. Histogram

A histogram depicts the distribution of pixel values within an image. The more evenly an image's pixel values are distributed, the more resistant it is to statistical analysis. The histograms of the three images and their corresponding cipher images are illustrated in Figure 16. The figure illustrates that the pixel value distribution of the three channels in the original image is nonuniform, whereas the pixel values of the encrypted image are distributed in an even range spanning from 0 to 255. This demonstrates that the encrypted image conceals any statistical features in the image, demonstrating the algorithm's resistance to statistical attacks.



(**a**) (**b**) (**c**) (**d**)

Figure 16. Histogram of original image and encrypted image: (**a**) image; (**b**) red histogram of (**a**); (**c**) green histogram of (**a**); (**d**) blue histogram of (**a**).

5.4. Correlation Analysis

The statistical assault is a method of attack that is relatively common in information transmission. To resist statistical attacks, an excellent image encryption algorithm must have a low relationship between adjacent images. The correlation coefficient reflects the relationship between pixels in an image adjacent horizontally, vertically, and obliquely. Following is the formula for calculating the correlation coefficient:

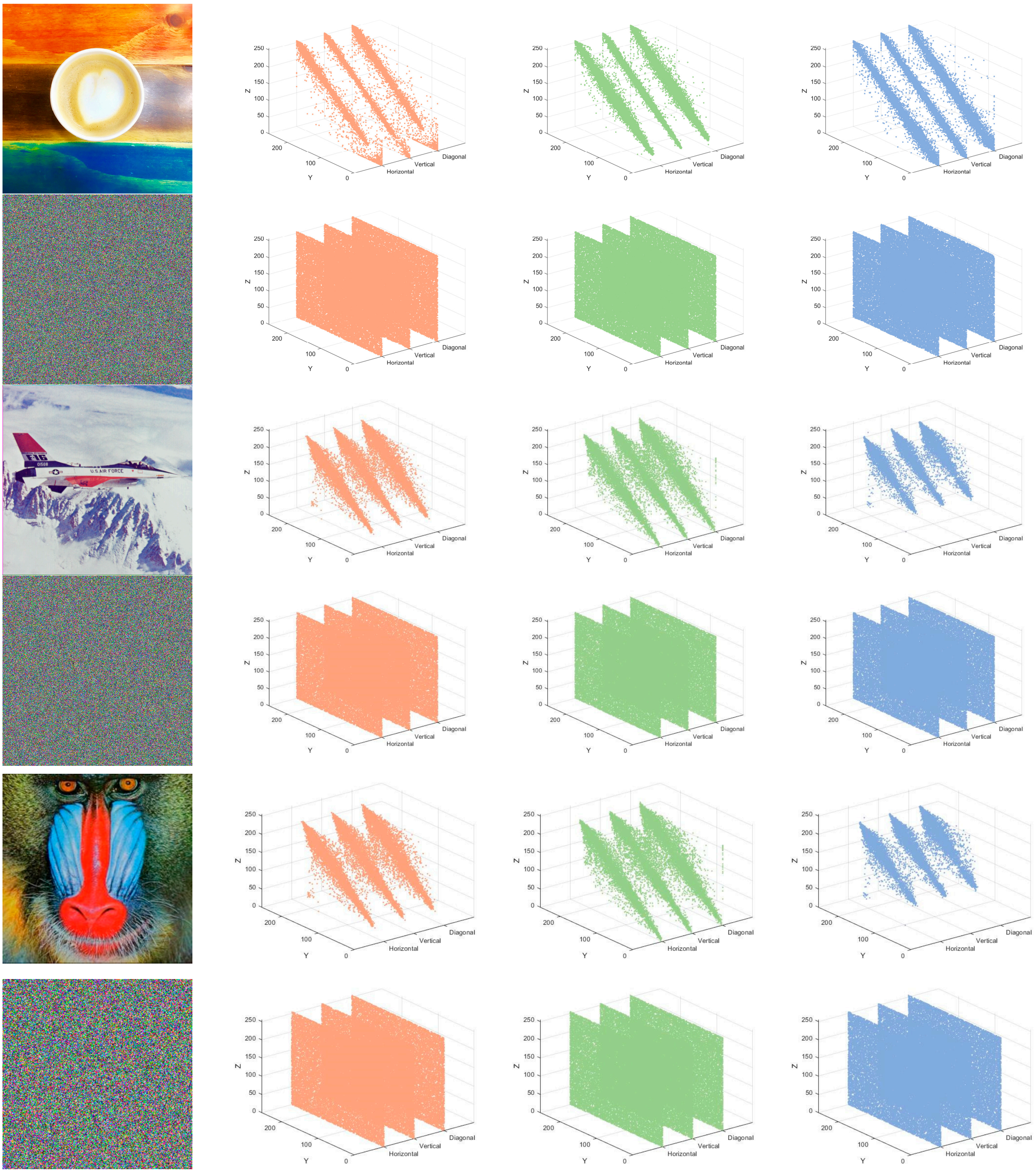
$$E(x) = \frac{1}{N}\sum\nolimits_{i=1}^{N}(x_i), \tag{16}$$

$$y(t+1) = ky(t) - z(t)f(t) - h, \tag{17}$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x_i))(y_i - E(y_i)), \tag{18}$$

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{19}$$

The correlation coefficient is positively correlated with the correlation between adjacent pixels. The greater a pixel's correlation, the closer its coefficient is to 1. Therefore, in order to ensure the security of the compiled image, its horizontal, vertical, and diagonal correlation coefficients should be as low as possible, indicating that no correlation exists between adjacent pixels. Figure 17 depicts the correlation between the original image and its ciphered image. Table 6 compares the correlation coefficients between original and encrypted images.



(**a**) (**b**) (**c**) (**d**)

Figure 17. Correlation simulation results: (**a**) image; (**b**) red plane of (**a**); (**c**) green plane of (**a**); (**d**) blue plane of (**a**).

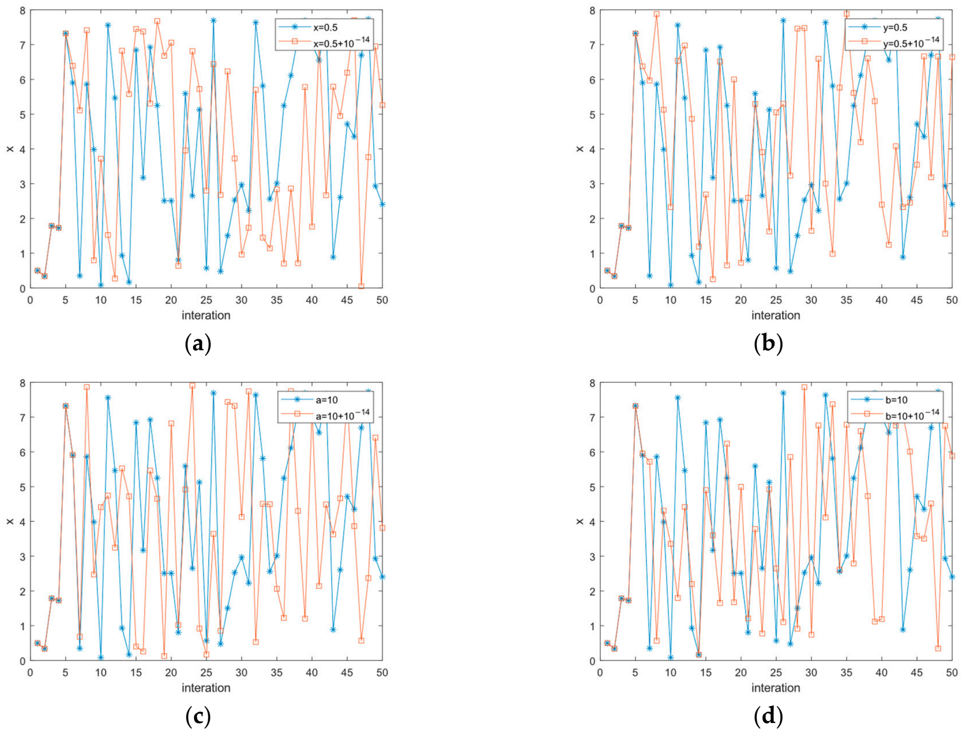
Table 6. Image correlation analysis.

Image	Image Size	Channel	File	Horizontal	Vertical	Diagonal
Lena	512 × 512 × 3	R	Original	0.97716	0.98774	0.96407
			Encrypted	−0.003545	0.0034381	−0.000291
			[11]	−0.0367	−0.0059	0.0182
			[31]	−0.0022	0.0057	0.00007
		G	Original	0.97731	0.98834	0.96433
			This paper	−0.0094703	0.0050664	0.0026209
			[11]	0.0030	0.0587	−0.0123
			[31]	0.0009	−0.0041	0.0038
		B	Original	09556	0.97433	0.93203
			This paper	−0.0042469	−0.0016389	0.000638
			[11]	−0.0037	−0.0227	−0.0134
			[31]	0.0013	0.0017	0.0104
Airplane	512 × 512 × 3	R	Original	0.97247	0.95675	0.93559
		G	This paper	0.001480	0.0069754	0.0061268
			Original	0.95786	0.96793	0.93352
		B	This paper	−0.0089081	−0.0014546	−0.0035603
			Original	0.96208	0.93595	0.91541
			This paper	0.0059152	−0.010841	0.0043028
Baboon	256 × 256 × 3	R	Original	0.94899	0.93012	0.90883
			This paper	−0.0078139	0.0018101	0.013868
			[27]	0.0078	0.0045	0.0765
			[32]	−0.0036	−0.0109	−0.0052
		G	Original	0.90897	0.8785	0.83944
			This paper	−0.00039714	−0.0055859	−0.0006205
			[27]	−0.0067	−0.0004	0.0078
			[32]	−0.0008	−0.0070	0.0095
		B	Original	0.94745	0.9339	0.90945
			This paper	0.011955	−0.017032	0.011271
			[27]	0.0045	−0.0023	0.0189
			[32]	−0.0009	0.0082	−0.0113

Figure 17 reveals that the correlation between the pixels of the ordinary image is extremely high, whereas the correlation between the pixels of the password image is evidently diminished. Table 6's results also support this conclusion. Strong correlation exists between adjacent pixels in the original image, so its correlation coefficient is close to 1. However, the correlation coefficient of the compiled image decreases considerably, and its average value is less than 0.01, demonstrating that there is currently no correlation between adjacent pixels. We believe that the algorithm presented in this study will significantly reduce the correlation between pixels for the purpose to withstand statistical attacks.

5.5. Key Sensitivity in Encryption

An effective encryption algorithm must be sensitive to the key, such that a minor change in the key will result in a substantial change in the output. The greater the key sensitivity, the larger the key space, and the greater its resistance to brute force attacks. This article's key is produced by hyperchaotic mapping. Therefore, the sensitivity of the key is also the sensitivity of the map. Figure 18 depicts the main difference produced by the four parameters x, y, a, and b in the algorithm described in this paper when they are altered slightly. After the fifth iteration, the chaotic system's key generation displays a significant dispersion, which suggests that the system is more affected by the initial value of the generated key.



(**a**) (**b**) (**c**) (**d**)

Figure 18. Initial value sensitivity analysis: (**a**) x = 0.5; (**b**) y = 0.5; (**c**) a = 10; (**d**) b = 10.

5.6. Key Space Analysis

Exhaustive attack is the most common attack method of various encryption algorithms. Therefore, whether it has good resistance to exhaustive attacks is an important indicator for evaluating the feasibility of an encryption algorithm. The key space will directly affect the algorithm's resistance to exhaustive attacks. The larger the key space, the more difficult it is to extract the key; that is, the stronger the security of the encryption algorithm. The key in this article consists of a time series of chaotic neurons, but different key sequences will be generated when the same time series is subjected to different modulo operations. Therefore, the key space of this encryption algorithm consists of 4 parameters. The range of its initial value is y and x ∈ (0, 255), a and b ∈ (0, +∞). At the same time, because double precision float can be accurate to 14 decimal places, the key space $K = 2^8 \times 10^{14} \times 2^8 \times 10^{14} \times 2^{63\times 2} \approx 2^{226}$. Its key space is much larger than 2^{120} [33]. Therefore, the key can resist brute force attacks.

5.7. Computational Complexity Analysis

The cryptographic efficiency standard is an important criterion for judging the quality of encryption algorithms. Specifically, it refers to the factors that affect the performance of encryption algorithms while ensuring security. Regarding cryptographic efficiency standards, the computational complexity of image encryption is the most critical indicator for evaluation. Computational complexity is related to the feasibility of the encryption algorithm and its requirements for hardware, which is mainly divided into time complexity and space complexity. The encryption algorithm in this paper mainly includes three stages: key generation, clock diffusion, and genetic recombination. In the process of key generation, we iterate the chaotic sequence M × N × 3 × 4 times. Clock diffusion and genetic recombination require one operation and M × N × 3 operations, respectively. Therefore, the total number of operations is 15 × M × N + 1, and its time complexity is O (MN). Table 7 shows the encryption and decryption time and comparison of the image in this study's algorithm. The time in the table is the average time of 10 encryption and decryption of the image. Since the gene recombination algorithm performs scrambling and diffusion at the same time, compared with existing encryption algorithms such as

DNA, we have eliminated the intermediate encoding process, so the encryption time has been significantly shortened, as shown in the table. The consistent encryption time of approximately 0.3 s for 512 × 512 × 3 demonstrates its effective encryption capability.

Table 7. Time comparison of different image encryption algorithms (Lena).

Algorithm		**256 × 256**	**512 × 512**
This article	Encryption	0.1358	0.3380
[34]	Encryption	0.5726	2.0181
[35]	Encryption	-	2.236
[36]	Encryption	2.2234	9.0013
[37]	Encryption	-	0.4842
[16]	Encryption	-	0.8385
[17]	Encryption	0.1119	-
This article	Decryption	0.1025	0.4271

At the same time, the space complexity of the encryption algorithm is an important criterion for whether the encryption algorithm can be truly applied, and it has certain requirements on the computer hardware. If an encryption algorithm has a high space complexity, the encryption process may be interrupted due to insufficient hardware memory during operation. In the encryption process of this paper, most of the calculation processes are direct operations on the original image, thus saving some intermediate storage. At the same time, for color images, only a single channel M × N length key is required to complete its encryption. In addition to the key requirements for genetic recombination, the encryption algorithm requires a key length of 10 × M × N, and its space complexity is O (MN). We compared it with the existing encryption algorithms, as shown in Table 8. We can find that, compared with the existing algorithms, the space complexity of this algorithm is basically consistent with them. However, compared with the quantum algorithm [38], its algorithm is not associated with the key but is related to the number of bits n after quantum encoding. Therefore, the space complexity of the algorithm is still a certain distance away from it.

Table 8. Space complexity.

Algorithm	**This Article**	**[39]**	**[37]**	**[33]**
space complexity	O (MN)	O (MN)	O (MN)	O (n)

5.8. Differential Attack Analysis

The purpose of a differential attack is to infer the possible value of the key by comparing the difference between a pair of plaintext and ciphertext. A qualified encryption algorithm should effectively resist differential attacks. Number of pixels rate of change (NPCR) and uniform change intensity (UACI) can be used to quantify this capability. The formulas for both are as follows:

$$NPCR = \frac{\underset{i,j}{\Sigma} D(i,j)}{M \times N} \times 100\%, \tag{20}$$

$$UACL = \frac{1}{M \times N}\left[\sum_{i,j} |C(i,j) - C\prime(i,j)|\right] \times 100\%, \tag{21}$$

M and N represent the image’s length and width, respectively, whereas C and C′ represent the encrypted image that is one pixel different from the original. The ideal expected values of NPCR and UACI were 99.6094% and 33.4635%, respectively. Wu [38] and others noted that the values of NPCR and UACI are dependent on the image format and image dimensions and are not static. NPCR should exceed 99.5693 and UACI should be at (33.2824, 33.6447) for a 256 × 256 image. NPCR should be greater than 99.5893, and UACI should be between (33.3730, 33.5541) for 512 × 512 images. Table 9 displays the pertinent outcomes and comparisons. The table demonstrates that all images of the algorithm presented in this paper meet the corresponding standards, indicating that the algorithm has a greater capacity to resist differential attacks.

Table 9. NPCR and UACL of encrypted images.

Image	Image Size	Algorithms	Channel	NPCR	UACL
Lena	512 × 512 × 3	This paper	R	99.6101	33.4234
		[40]		99.6535	33.4943
		[37]		99.6140	33.5627
		[41]		99.6136	33.4783
		This paper	G	99.6063	33.5112
		[40]		99.5770	33.5117
		[37]		99.6017	33.5218
		[41]		99.5922	33.4796
		This paper	B	99.6014	33.5513
		[40]		99.6560	33.5901
		[37]		99.6140	33.4339
		[41]		99.6109	33.4916
Airplane	512 × 512 × 3	This paper	R	99.5880	33.4633
		This paper	G	99.6143	33.4958
		This paper	B	99.6273	33.4626
Baboon	256 × 256 × 3	This paper	R	99.6368	33.4628
		This paper	G	99.5834	33.4397
		This paper	B	99.5987	33.5125

5.9. Image Quality Analysis

When evaluating the encryption algorithm, the image quality [42] evaluation will be used as one of the common indicators; the encrypted image will be compared to the original image after decryption in order to determine if the information has been lost during the encryption process. PSNR is a standard metric used to evaluate image quality. This index determines whether information has been lost by comparing the signal-to-noise ratio of the decrypted and original images. The formula is as follows:

$$PSNR = 10 \times \log_{10} \frac{(MAX)^2}{MSE}, \tag{22}$$

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (X(i,j) - Y(i,j))^2, \tag{23}$$

Among them, MAX represents the utmost pixel value, which is 255. Mean square error (MSE) is the sum of the squares of the pixel differences between two images; it can also be used as a measure of image quality. Generally speaking, if the value of PSNR is between (20, 30), it can be considered that the quality of the image is high, and the final decrypted image and the original image do not have a large loss of information. The results of the assault tests described in Sections 5.10 and 5.11 will be presented and divided using PSNR as an indicator, as shown in Table 10.

5.10. Resistant to Noise Attacks

In the process of transmitting image data, in addition to various malicious assaults, environmental interference is a significant cause of information loss. Communication noise is a type of environmental interference that is more likely to occur during the image transmission process. Certain information will be lost after decryption if the image has been corrupted by noise, and it may even be impossible to decrypt the image. Gaussian noise and salt-and-pepper noise are chosen for simulation experiments on airplane (512 × 512) in this paper. Gaussian noise with a mean value of 0 and a variance of 0.0005, 0.005, 0.01, and salt-and-pepper noise of 5% are selected to interfere with the encrypted image. The comparison diagram following decryption is depicted in Figure 19.

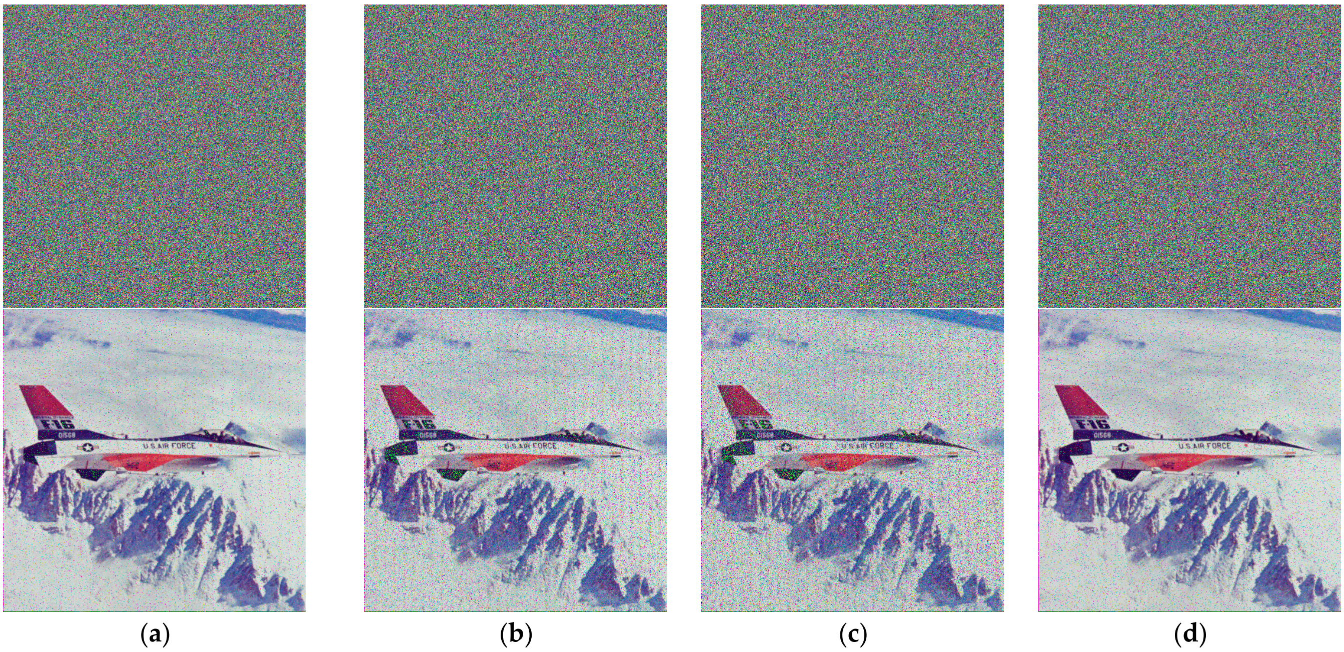


Figure 19. Gaussian noise: (**a**) 0.0005; (**b**) 0.005; (**c**) 0.01; (**d**) salt-and-pepper noise.

Visually, the decrypted image after noise pollution will generate a certain amount of noise, but it does not lose a significant amount of the original image's information, which is acceptable. Table 9 displays concurrently the PSNR value of the decrypted image. Under the influence of Gaussian noise, the PSNR value is still greater than 20, and the image quality is acceptable. Although the PSNR values under the influence of salt-and-pepper noise are below 20, they are all greater than 18. Experimental results indicate that the algorithm has some noise pollution resistance.

5.11. Missing Data

In addition to the possibility of noise contamination during Internet image transmission, data loss is another reason to reuse image information. During transmission, the image may encounter network congestion, data packet loss, or flaws in the data transmission protocol, resulting in the terminal receiving incomplete data. Two categories of image data loss exist: blocking and cropping attacks. The former is the loss of a portion of a single channel, while the latter is the simultaneous loss of all pixel values in an area. In this paper, simulation experiments are conducted on both types. After encryption, the blocking attack causes the airplane image to lose varying channel values in various regions. The cropping attack will remove 6.25 percent, 12.5 percent, 25 percent, and 50 percent of the encrypted airplane image. The comparison between Figures 20 and 21 depicts decryption.

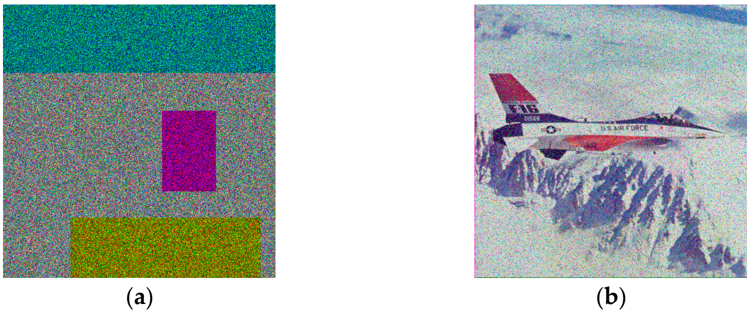


Figure 20. Blocking attack. (**a**) Picture after channel loss; (**b**) Decrypted image after the attack.

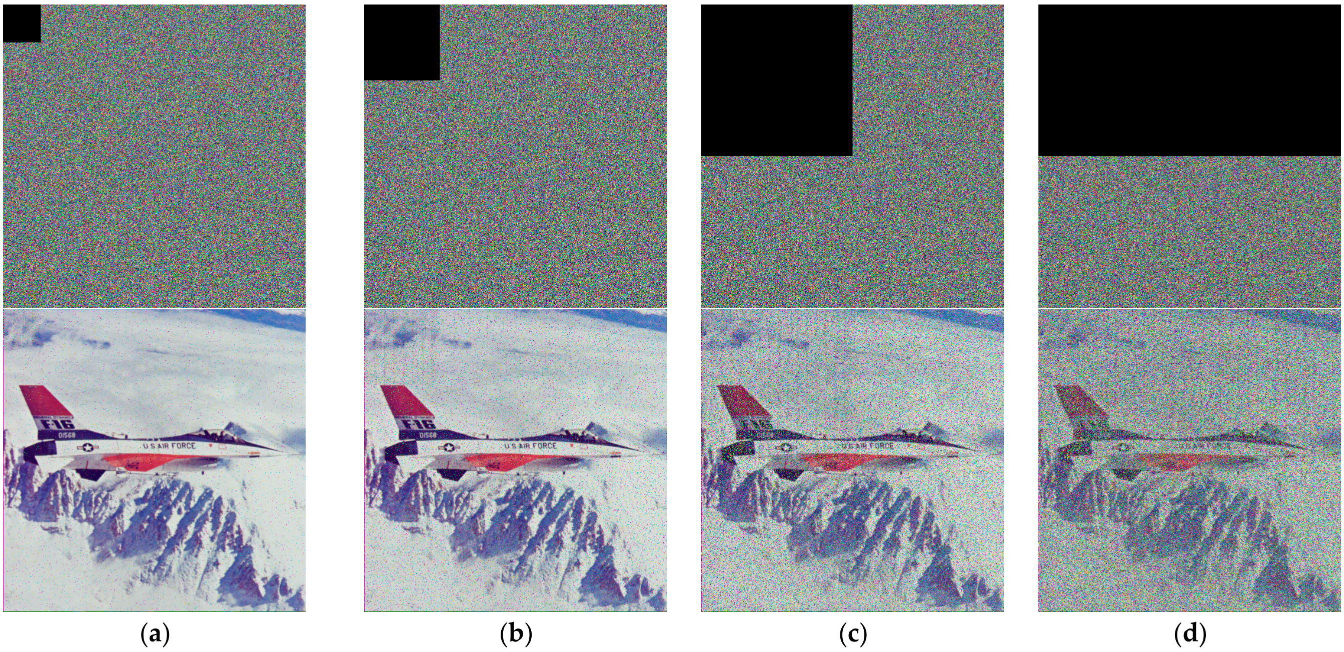


Figure 21. Blocking attack: (**a**) 6.25% shear; (**b**) 12.5% shear; (**c**) 25% shear; (**d**) 50% shear.

Although the blocking attack results in the loss of pixel values in different areas of multiple channels, the majority of the original image's information is still visible, whereas the cropping attack does not change considerably until 50% of the image is lost. When 50% of the data is lost, the image becomes evidently blurry, but the original image's outline and content can still be seen. Table 10 shows the PSNR of the experimental results. In the blocking attack, it is evident that the PSNR of the G channel with the greatest loss is considerably lower than that of the other two channels, but its average value is still greater than 18. As the loss area increases in the clipping attack, the PSNR begins to decline sharply, but the average PSNR is still above 10, even in the 50 lost area. It can be seen that the algorithm confuses the pixel sequence while encrypting the pixel value, and that it is more resistant to data loss attacks.

Table 10. PSNR result.

Attack	R Channel	G Channel	B Channel
Gauss noise—0.0005	21.0713	20.4366	21.6251
Gauss noise—0.005	16.2561	15.5950	16.6946
Gauss noise—0.01	14.3329	13.6317	14.4615
Salt and pepper noise	20.1299	19.9205	20.1488
Block attack	15.6317	15.5994	15.8552
6.25% shearing	25.7698	25.5021	25.6311
12.5% shearing	19.6441	19.4130	19.7270
25% shearing	13.6640	13.4004	13.7402
50% shearing	10.8459	10.5556	10.7402

6. Conclusions

This study studied the Ackley function and the Styblinski–Tang function and cross-mixed the two to acquire a new type of two-dimensional hyperchaotic mapping, 2D-AST mapping. Through the utilization of various indicators, including the LE, information entropy, permutation entropy, phase space trajectory, and bifurcation diagram, it has been demonstrated that the hyperchaotic mapping encompasses a more extensive range of chaos states than the current chaotic mapping. Moreover, it enables the chaotic dynamics system to maintain its chaotic state with greater stability. As a result, NIST randomness testing was performed on the time series produced by 2D-AST to ensure that they are suitable for implementation in image encryption algorithms. This paper concurrently presents algorithms for genetic recombination and clock diffusion. Clock diffusion simulates the second hand, minute hand, and hour hand of the three channels of the color image, thereby increasing the efficacy of the encryption algorithm by diffusing all three channels. Simultaneously, we chose two distinct data structures, queue and stack, from the existing data structures to construct a new composite data structure. By standardizing the data structure regulations, we integrate the mechanisms of gene chain fragmentation and recombination seen in nature, thus introducing a novel image encryption methodology. The color image encryption algorithm, which incorporates clock diffusion and gene recombination and generates keys via 2D-AST chaotic mapping, has been evaluated against conventional encryption techniques. The information entropy of the 512 × 512 × 3 image remains stable at 7.9998, and that of the 256 × 256 × 3 image is stable at 7.9990. Moreover, compared to existing multi-class diffusion algorithms, its encryption efficiency has been significantly enhanced while maintaining the encryption effect. Averaging 0.3 s for 512 × 512 color images and 0.1 s for 256 × 256 color images. This demonstrates that while maintaining image quality, the color image encryption algorithm circumvents the lengthy encoding and decoding process of the diffusion algorithm. The viability of the gene recombination encryption technique has enhanced the diversity of image data representation, hence facilitating the further development of image encryption algorithms.

However, we also found that the image encryption algorithm has certain limitations when encrypting high-precision images, and the length of its key is positively correlated with the size of the image. When encrypting high-resolution images, the required key length also gets longer and longer, which puts higher requirements on computer hardware. Therefore, how to reduce the storage space required for chaos without affecting chaos degradation will be our future research direction.

Author Contributions: Y.X. gave some theoretical guidance; J.L. completed the experiment simulation and wrote the paper; Z.Y. and T.Z. gave guidance on writing the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Heilongjiang Province, grant number LH2021F035 and Harbin University of Commerce Teacher Innovation Project Support Plan 2021.

Data Availability Statement: Dataset available on request from the authors.

Acknowledgments: We thank the editors and the reviewers for their constructive suggestions and insightful comments, which helped us greatly to improve this manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Shannon, E. Communication theory of secrecy systems. *Bell Syst. Tech. J.* **1949**, *28*, 656–715. [CrossRef]
2. Aihara, K.; Takabe, T.; Toyoda, M. Chaotic neural networks. *Phys. Lett. A* **1990**, *144*, 333–340. [CrossRef]
3. Zhou, Y.; Bao, L.; Chen CL, P. Image encryption using a new parametric switching chaotic system. *Signal Process.* **2013**, *93*, 3039–3052. [CrossRef]
4. Pak, C.; Huang, L. A new color image encryption using combination of the 1D chaotic map. *Signal Process.* **2017**, *138*, 129–137. [CrossRef]
5. Natiq, H.; Al-Saidi, N.M.G.; Said, M.R.M.; Kilicman, A. A new hyperchaotic map and its application for image encryption. *Eur. Phys. J. Plus* **2018**, *133*, 1–14. [CrossRef]
6. Cao, C.; Sun, K.; Liu, W. A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map. *Signal Process.* **2018**, *143*, 122–133. [CrossRef]
7. Toktas, F.; Erkan, U.; Yetgin, Z. Cross-channel color image encryption through 2D hyperchaotic hybrid map of optimization test functions. *Expert Syst. Appl.* **2024**, *249*, 123583. [CrossRef]
8. Chen, G.; Mao, Y.; Chui, K. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Solitons Fractals* **2004**, *21*, 749–761. [CrossRef]

9. Wang, X.Y.; Yang, L.; Liu, R.; Kadir, A. A chaotic image encryption algorithm based on perceptron model. *Nonlinear Dyn.* **2010**, *62*, 615–621. [CrossRef]
10. Wang, X.; Zhang, H. A novel image encryption algorithm based on genetic recombination and hyper-chaotic systems. *Nonlinear Dyn.* **2016**, *83*, 333–346. [CrossRef]
11. Xu, Y.; Tang, M. Color image encryption algorithm using dna encoding and fuzzy single neurons. *IEEE Access* **2022**, *10*, 127770–127782. [CrossRef]
12. Zhou, G.; Sun, Y.J.; Fan, P. Quantum image Gray-code and bit-plane scrambling. *Quantum Inf. Process.* **2015**, *1*, 1717–1734. [CrossRef]
13. Teng, L.; Wang, X.; Yang, F.; Xian, Y. Color image encryption based on cross 2D hyperchaotic map using combined cycle shift scrambling and selecting diffusion. *Nonlinear Dyn.* **2021**, *105*, 1859–1876. [CrossRef]
14. Sun, J. 2D-SCMCI hyperchaotic map for image encryption algorithm. *IEEE Access* **2021**, *9*, 59313–59327. [CrossRef]
15. Hua, Z.; Zhu, Z.; Chen, Y.; Li, Y. Color image encryption using orthogonal Latin squares and a new 2D chaotic system. *Nonlinear Dyn.* **2021**, *104*, 4505–4522. [CrossRef]
16. Qiu, H.; Xu, X.; Jiang, Z.; Sun, K.; Xiao, C. A color image encryption algorithm based on hyperchaotic map and Rubik's Cube scrambling. *Nonlinear Dyn.* **2022**, *110*, 2869–2887. [CrossRef]
17. Wang, Q.; Zhang, X.; Zhao, X. Color image encryption algorithm based on novel 2d hyper-chaotic system and dna crossover and mutation. *Nonlinear Dyn.* **2023**, *111*, 22679–22705. [CrossRef]
18. Wang, X.; Chen, X.; Zhao, M. A new two-dimensional sine-coupled-logistic map and its application in image encryption. *Multimed. Tools Appl.* **2023**, *82*, 35719–35755. [CrossRef]
19. Ackley, D. *A Connectionist Machine for Genetic Hillclimbing*; Springer Science & Business Media: Berlin, Germany, 2012; pp. 13–14.
20. Styblinski, M.A.; Tang, T.S. Experiments in nonconvex optimization: Stochastic approximation with function smoothing and simulated annealing. *Neural Netw.* **1990**, *3*, 467–483. [CrossRef]
21. Xu, X.; Chen, S. Single neuronal dynamical system in self-feedbacked Hopfield networks and its application in image encryption. *Entropy* **2021**, *23*, 456. [CrossRef]
22. Richman, J.S.; Moorman, J.R. Physiological time-series analysis using approximate entropy and sample entropy. *Am. J. Physiol. -Heart Circ. Physiol.* **2000**, *278*, H2039–H2049. [CrossRef] [PubMed]
23. Xu, X.; Chen, S. A remote sensing image encryption method combining chaotic neuron and tent map. *J. Comput.* **2021**, *32*, 108–123.
24. Bandt, C.; Pompe, B. Permutation entropy: A natural complexity measure for time series. *Phys. Rev. Lett.* **2002**, *88*, 174102. [CrossRef]
25. Toktas, A.; Erkan, U.; Gao, S.; Pak, C. A robust bit-level image encryption based on Bessel map. *Appl. Math. Comput.* **2024**, *462*, 128340. [CrossRef]
26. Osman, F. Double-strand break-induced recombination in eukaryotes. *Prog. Nucleic Acid Res. Mol. Biol.* **1997**, *58*, 263–299.
27. Mfungo, D.E.; Fu, X. Fractal-Based Hybrid Cryptosystem: Enhancing Image Encryption with RSA, Homomorphic Encryption, and Chaotic Maps. *Entropy* **2023**, *25*, 1478. [CrossRef]
28. Wang, S.; Sun, B.; Wang, Y.; Du, B. Image encryption algorithm using multi-base diffusion and a new four-dimensional chaotic system. *Multimed. Tools Appl.* **2024**, *83*, 10039–10060. [CrossRef]
29. Khalil, N.; Sarhan, A.; Alshewimy, M.A.M. An efficient color/grayscale image encryption scheme based on hybrid chaotic maps. *Opt. Laser Technol.* **2021**, *143*, 107326. [CrossRef]
30. Li, X.; Sun, B.; Bi, X.; Yan, H.; Wang, L. A Novel Color Image Encryption Algorithm Based on Cross-plane Scrambling and Diffusion. *Mob. Netw. Appl.* **2023**, 1–12. [CrossRef]
31. Lim, Z.; Peng, C.; Tan, W.; Li, L. A novel chaos-based color image encryption scheme using bit-level permutation. *Symmetry* **2020**, *12*, 1497. [CrossRef]
32. Wang, Q.; Zhang, X.; Zhao, X. Color image encryption algorithm based on bidirectional spiral transformation and DNA coding. *Phys. Scr.* **2023**, *98*, 025211. [CrossRef]
33. He, J.; Zhu, H.; Zhou, X. Quantum image encryption algorithm via optimized quantum circuit and parity bit-plane permutation. *J. Inf. Secur. Appl.* **2024**, *81*, 103698. [CrossRef]
34. Wang, X.; Zhang, X.; Gao, M.; Tian, Y.; Wang, C.; Iu, H.H.C. A color image encryption algorithm based on hash table, hilbert curve and hyper-chaotic synchronization. *Mathematics* **2023**, *11*, 567. [CrossRef]
35. Xu, Y.Q.; Zhen, X.X.; Tang, M. Dynamical system in chaotic neurons with time delay self-feedback and its application in color image encryption. *Complexity* **2022**, *2022*, 2832104. [CrossRef]
36. Li, P.; Zhang, X. Image encryption algorithm based on a novel cascade chaotic system and DNA mutation. *Phys. Scr.* **2024**, *99*, 105203. [CrossRef]
37. Wang, M.; Fu, X.; Yan, X.; Teng, L. A New Chaos-Based Image Encryption Algorithm Based on Discrete Fourier Transform and Improved Joseph Traversal. *Mathematics* **2024**, *12*, 638. [CrossRef]
38. Wu, Y.; Noonan, J.P.; Agaian, S. NPCR and UACI randomness tests for image encryption. *Cyber J. Multidiscip. J. Sci. Technol. J. Sel. Areas Telecommun. (JSAT)* **2011**, *1*, 31–38.
39. Alawida, M.A. Novel DNA tree-based chaotic image encryption algorithm. *J. Inf. Secur. Appl.* **2024**, *83*, 103791. [CrossRef]
40. Wang, X.; Su, Y.; Luo, C.; Nian, F.; Teng, L. Color image encryption algorithm based on hyperchaotic system and improved quantum revolving gate. *Multimed. Tools Appl.* **2022**, *81*, 13845–13865. [CrossRef]

41. Xu, J.; Zhao, B.; Wu, Z. Research on color image encryption algorithm based on bit-plane and Chen Chaotic System. *Entropy* **2022**, *24*, 186. [CrossRef]
42. Xu, X.; Chen, S. An optical image encryption method using Hopfield neural network. *Entropy* **2022**, *24*, 521. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.